UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Abordagem Imunológica de Segurança Baseada em Correlação de Alertas e Redes Programáveis

Roberto Vasconcelos Melo

São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Roberto Vasconcelos Melo

Abordagem Imunológica de Segurança Baseada em Correlação de Alertas e Redes Programáveis

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação (PROCC) da Universidade Federal de Sergipe (UFS) como requisito para a obtenção do título de mestre em Ciência da Computação.

Orientador: Prof. Dr. Douglas Dyllon Jeronimo de Macedo

São Cristóvão – Sergipe

2018

Resumo

Na área de segurança, técnicas de detecção de anomalia foram desenvolvidas com o objetivo de detectar padrões de tráfego que representam ataques ou atividades maliciosas e são frequentemente referidos como anomalias. Particularmente, algumas anomalias podem estar associadas a invasores que executam ataques de negação de serviço distribuído (*Distributed Denial-of-Service* - DDoS) para degradar a disponibilidade de serviços *online*. Ameaças na categoria DoS podem envolver estágios iniciais, como ataques de reconhecimento. Nesse tipo de ameaça, a rede é escaneada com o objetivo de encontrar máquinas vulneráveis e comprometê-las. Dessa forma, as vulnerabilidades detectadas possibilitam o acesso não autorizado às máquinas por meio de ataques nas classes de usuário para super usuário (U2R) e remoto para local (R2L). As máquinas comprometidas podem ser utilizadas com o intuito de provocar a negação de serviço contra determinado alvo. Essas classes contém ataques que podem se esconder no tráfego normal devido à baixa intensidade de ataque requerida. Além disso, as técnicas de detecção baseadas em anomalia apresentam uma alta taxa de alarmes falsos, o que prejudica a eficácia da detecção.

Para atenuar esses problemas, o presente trabalho tem como objetivo apresentar uma abordagem de segurança com a função de detectar e mitigar ataques que exploram vulnerabilidades da infraestrutura da nuvem. Essa abordagem consiste nos conceitos de imunologia, correlação de alertas e redes programáveis. A partir dela, um sistema de detecção de intrusão baseado em anomalia, e dentro da abordagem imunológica, é utilizado em conjunto com uma técnica de correlação de alertas baseada em grafos de ataque. Neste trabalho, os algoritmos de seleção negativa, seleção clonal e rede imune são usados para implementar um sistema de detecção baseado em agentes distribuídos para analisar o tráfego de rede. O sistema descrito é utilizado com o auxílio de grafos de ataque, a partir do qual um algoritmo de correlação de alertas pode auxiliar na taxa de redução de alarmes falsos. Grafos de Ataque podem também auxiliar na seleção de contramedidas baseadas na tecnologia de redes programáveis (SDN - *Software Defined Networks*), a partir da qual podem ser executadas medidas de prevenção como redirecionamento, ou isolamento do tráfego, variação na topologia da rede, e mudanças de endereços IP.

A abordagem proposta foi testada a partir do tráfego de rede coletado das máquinas virtuais do *Amazon Web Service* (AWS), onde para sua análise ele foi convertido para *datasets* no formato NSL-KDD. A adição da técnica de correlação aumentou a eficácia da detecção para todas as classes de ataques estudadas.

Palavras-chave: AIS, IDS, IPS, Comutação em Nuvem, Redes SDN.

Abstract

In the security field, anomaly detection techniques have been developed to detect traffic patterns related to attacks or malicious activities and are often referred to as anomalies. Particularly, some anomalies can represent attackers launching Distributed Denial-of-Service (DDoS) in order to degrade services availability. Threats in the DoS category can involve early-stage actions such as probe attacks. In this type of attack, a network is scanned in order to find vulnerable hosts and compromise them. As a result, the detected vulnerabilities enable unauthorized access to the machines through user to root (U2R) and remote to local (R2L) attacks. The compromised machines could be used in order to cause a denial of service against a particular target. These attack classes include threats that can hide in normal traffic due to low required attack intensity. In addition, anomaly-based detection techniques have a high false alarm rate, which helps in reducing the detection efficiency.

This work aims to present a security model with detection and prevention functions against attacks that exploit the vulnerabilities of the cloud infrastructure to mitigate the previously mentioned problems. This model consists of the concepts of immunology, alert correlation, and software-defined networks (SDN). It consists of a distributed intrusion detection system based on anomaly detection within the artificial immune system (AIS) approach and attack graph correlation. Through this approach, an anomaly-based intrusion detection system inside the AIS field works with attack graph based correlation. The Negative Selection, Clonal Selection and Immune Network algorithms are used to implement an agent-based detection system to analyze network traffic. The described system works in conjunction with attack graphs and an alert correlation algorithm which can aid in the false alarm reduction rate. Attack graphs can also aid in the countermeasure selection through SDN technology. The SDN countermeasures can assist in attack prevention through traffic redirection, traffic isolation, network topology change, and IP address change.

The proposed system was tested through the network traffic collected from the virtual machines on Amazon Web Service (AWS). The collected traffic data was converted to datasets in the NSL-KDD format. The addition of alert correlation technique in the proposed security approach increased detection efficiency for all studied attack classes.

Keywords: AIS, IDS, IPS, Cloud Computing, SDN Networks.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ACD	Algoritmo de Células Dendríticas
CD	Células Dendríticas
MCAV	Valor de Antígeno de Contexto Maduro
NSA	Algoritmo de Seleção Negativa
CSA	Algoritmo de Seleção Clonal
INA	Algoritmo de Rede Imune
IDS	Sistema de Detecção de Intrusão
NIDS	Sistema de Detecção de Intrusão de Redes
IPS	Sistema de Prevenção de Intrusão
AIS	Sistema Imunológico Artificial
HIS	Sistema Imunológico Humano
IaaS	Infraestrutura como Serviço
PaaS	Plataforma como Serviço
SaaS	*Software* como serviço
VM	Máquina Virtual
DDoS	Ataque de Negação de Serviço Distribuído
R2L	Remoto para Local
U2R	Usuário para Super Usuário
SAG	Grafo de Cenário de Ataque
ACG	Grafo de Correlação de Alertas
AGC	Correlação de Alertas baseado em Grafos de Ataque
DAG	Gráfico Acíclico Dirigido
BN	Rede Bayesiana
DPI	Inspeção Profunda de Pacotes

SDN	Rede Definida por *Software*
CVSS	*Common Vulnerability Scoring System*
VSI	Índice de Segurança de VM

Sumário

1 Introdução

Devido à popularização da computação nas décadas de 80 e 90, grandes empresas utilizando redes financeiras e sistemas de comunicação passaram a depender cada dia mais de sistemas computacionais. Consequentemente, começaram a surgir diversas ameaças virtuais. No sentido de trazer melhores níveis de segurança para esses sistemas, vários métodos e técnicas foram desenvolvidos. Entre eles podemos elencar: *firewalls*, IPS, Anti-Vírus, Anti-Spam. Como consequência da crescente preocupação na área de segurança os primeiros trabalhos relacionados à IDS (*Intrusion Detection Systems*) começaram a surgir na década de 80 (ANDERSON et al., 1980)(DENNING, 1987).

Os IDS podem se enquadrar em algumas categorias, dentre elas podemos citar: tipos de sistemas de detecção de intrusão, forma de detecção e modelo de utilização. O primeiro se refere ao ambiente que o IDS irá monitorar, caso monitore uma máquina, será considerado um IDS baseado em Host (HIDS – *Host based Intrusion Detection System*). Já no caso em que monitore uma rede será considerado um IDS baseado em rede. No entanto ainda existem os casos de IDS híbridos que podem monitorar ambos ambientes. A segunda categoria se foca nas formas de detecção de uma ameaça. Onde existe a forma de detecção por assinatura, em que o sistema possui informações prévias sobre padrões pré-definidos de ataques ou outras atividades maliciosas. Nessa técnica o sistema apenas reconhece ataques em que já possui informação prévia sobre eles. Já na detecção por anomalia é montado um perfil que representa o comportamento normal do usuário, sistema ou rede. No caso de alguma atividade se encontrar fora desse perfil, ela poderá ser considerada uma ameaça. Uma desvantagem dessa técnica se deve ao alto número de alarmes falsos. Por último temos o modo de utilização, que pode ser passivo ou reativo. No passivo o sistema apenas detecta a ameaça, enquanto no reativo além de detectar a ameaça, ele também tenta combatê-la (SISTEMAS. . . , 2010).

Existem diversos ambientes nos quais sistemas de detecção de intrusão podem ser implantados. De acordo com (SRIHARI; KALPANA; ANITHA, 2014), foi desenvolvido um sistema capaz de detectar ameaças do tipo Voice Spam em redes VoIP (*Voice over IP*). O objetivo desse trabalho é detectar e prevenir chamadas de voz não solicitadas (propagandas, enquetes telefônicas e telemarketing) em redes, onde a transmissão de voz ou de conteúdos de multimídia ocorrem por meio do protocolo IP. Já o trabalho de (KUMAR; REDDY, 2014) utiliza uma técnica baseada em agentes dentro da abordagem imunológica com o objetivo de identificar ameaças em redes sem fio, tanto a nível de pacote quanto a nível de sinal. Enquanto outros trabalhos buscam detectar intrusões em ambientes onde os recursos de memória e processamento são limitados. Um exemplo deste último seria o trabalho de (SHAMSHIRBAND et al., 2014) que busca detectar ataques de negação de serviço distribuído em redes de sensores sem fio. Esse tipo de ataque busca tornar os recursos de um sistema indisponíveis para seus utilizadores.

Um ambiente que requer uma maior preocupação referente à segurança é o da computação na nuvem, pois devido à sua natureza distribuída, ela possui mais vetores de ataques, o que a torna mais vulnerável (VIEIRA et al., 2010). Uma vez que ela é composta por um conjunto de servidores e máquinas virtuais (VM – *Virtual Machine*), as aplicações podem ser executadas de forma distribuída. Além disso, ela possui a característica de ser *multitenancy*, onde uma instância lógica de algum aplicativo pode ser compartilhada com diversos clientes. Sendo assim, surgem problemas relacionados à segurança e privacidade em que, por exemplo, usuários maliciosos podem explorar as vulnerabilidades desse ambiente, procurando obter acesso a dados privados e limitando o poder de processamento, largura de banda, ou capacidade de armazenamento de redes na nuvem (OKTAY; SAHINGOZ, 2013).

No ambiente da nuvem existem três modelos de serviço fornecidos, em que cada um deles possui preocupações relacionadas à segurança. O primeiro denomina-se SaaS (*Software as a Service*), onde é possível obter acesso a aplicativos instalados em um servidor remoto. Nesse modelo existem preocupações em relação à forma como os dados são armazenados e protegidos, uma vez que eles se encontram armazenados fora dos limites das empresas, resultando em um desinteresse por parte delas em adotar esse tipo de serviço (SUBASHINI; KAVITHA, 2011). Já o segundo modelo é conhecido como PaaS (*Platform as a Service*), em que é fornecido acesso a um ambiente de desenvolvimento remoto, possibilitando o desenvolvimento de projetos de software sem a necessidade de um framework instalado localmente. No entanto a abertura que esse serviço fornece em relação ao desenvolvimento de aplicativos utilizando plataformas remotas, também pode auxiliar usuários mal-intencionados a desenvolver aplicativos maliciosos que comprometam a infraestrutura da nuvem. Por essa razão é necessário implementar mecanismos de segurança confiáveis que se encontrem abaixo do nível dessas aplicações, como sistemas de prevenção de redes, por exemplo (SUBASHINI; KAVITHA, 2011). Por último existe o serviço IaaS (*Infrastructure as a Service*), no qual a infraestrutura da nuvem é virtualizada, onde através desse processo a sua infraestrutura pode ser utilizada por meio de máquinas virtuais, como por exemplo, para hospedar algum aplicativo de uma empresa (DESVENDANDO..., 2017). No entanto se houver vulnerabilidades no software de virtualização, usuários mal-intencionados poderão explorá-las, comprometendo todos os modelos de serviço da nuvem, uma vez que o modelo IaaS se trata da base para todos os outros tipos de serviços fornecidos (SUBASHINI; KAVITHA, 2011).

Uma abordagem que pode contribuir muito para tornar o ambiente da nuvem mais seguro se refere à redes SDN (*Software Defined Networking*). Essa abordagem está cada vez mais ganhando destaque na área de segurança, uma vez que são empregadas em conjunto com sistemas de detecção de intrusão para redes (NIDS - *Network Intrusion Detection System*) com o objetivo de mitigar danos que possam ser causados em determinado ambiente. Esse tipo de rede possui a característica especial de desacoplar a lógica de controle de implementações fechadas de proprietário, permitindo dessa forma que pesquisadores e proprietários possam desenvolver novas funções ou protocolos de forma mais fácil e flexível (SHIN; GU, 2013). Como consequência

já existem trabalhos bem citados na área de segurança que se utilizam desse recurso, como os frameworks NICE (*Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems*) (CHUNG et al., 2013) e SnortFlow (XING et al., 2013). Ambos se tratam de sistemas na área de segurança que pretendem tornar o ambiente da nuvem mais seguro por meio da abordagem SDN, para que seja possível selecionar medidas de segurança através da programação da rede, para minimizar danos que possam ser causados por algum ataque.

A área de ciência da computação possui a tradição de reproduzir ideias da natureza para a solução de problemas, onde uma delas se refere ao sistemas imunológicos artificiais (AIS - *Artificial Immune System*). Isso posto, o sistema imunológico humano (HIS – *Human Immune System*) é composto por células interdependentes que protegem o corpo humano contra diversas infecções patogênicas, como bactérias, vírus e parasitas. Onde o objetivo do HIS se trata de diferenciar o *self* (moléculas e células que pertencem ao organismo) do *non-self* (moléculas e células que são reconhecidas como estrangeiras) (YANG et al., 2014). Dessa forma é possível utilizar essa ideia para IDS com o objetivo de detectar intrusões, uma vez que se torna possível desenvolver sistemas de detecção baseados em anomalia e dentro da abordagem imunológica. O *self* seria representado pelo perfil normal do sistema, ou seja, ausência de intrusões, enquanto o *non-self* representaria um perfil anômalo, ou seja, a presença de intrusões. Sendo assim, essa abordagem possui quatro principais paradigmas que serão explicados no decorrer desse trabalho, que são teoria do perigo, rede imune, seleção negativa e seleção clonal.

Um problema recorrente em sistemas de detecção baseados em anomalia se deve a alta taxa de alarmes falsos. Entre as técnicas que buscam mitigar esse problema pode-se elencar correlação de alertas. Essa técnica consiste em agregar alertas gerados por um IDS para construir um cenário de ataque (HUBBALLI; SURYANARAYANAN, 2014). Isso se deve ao fato de que os alertas gerados possuem pouca informação sobre uma determinada ameaça. Por essa razão a correlação de alertas auxilia na análise de um determinado cenário para que se torne possível confirmar um ataque, ou realizar alguma medida de prevenção para mitiga-lo (HUBBALLI; SURYANARAYANAN, 2014). Entre os métodos de correlação existentes, pode-se citar a correlação baseada em grafos de ataque. Esses grafos representam um mapa de vulnerabilidades presente em uma rede a partir do qual os alertas serão mapeados e correlacionados (CHUNG et al., 2013).

Dessa maneira, esse trabalho visa apresentar uma abordagem de segurança para nuvens computacionais e redes convencionais, focado em combater quatro classes de ataques: Negação de Serviço, remoto para local (R2L - *Remote to Local*), usuário para super usuário (U2R - *User to Root*) e ataques de reconhecimento (*Probe Attacks*). A primeira classe possui o objetivo de tornar serviços indisponíveis para os seus usuários. O segundo grupo visa possuir acesso remoto e não autorizado a uma determinada máquina. O terceiro grupo, também conhecido como ataques de escala de privilégio, garante acesso de administrador ao sistema a usuários mal intencionados. Por último, existem ataques cujo objetivo é escanear uma rede para encontrar vulnerabilidades e,

por meio dessas informações, comprometê-la (AMBEDKAR; BABU, 2015).

Sendo assim, o modelo desenvolvido apresenta um NIDS em que a abordagem imunológica é utilizada para implementar um sistema de detecção de acordo com o trabalho de (*Multi-Agent Artificial Immune System IDS*) (SERESHT; AZMI, 2014). Esse sistema trabalha em conjunto com um algoritmo de correlação de alertas para redução de alarmes falsos. Dessa forma os alertas podem ser mapeados e correlacionados por meio de um grafo de ataque. A tecnologia de redes SDN é utilizada na execução de contramedidas que visam a mitigação dos ataques,

1.1 Problemática e Hipótese

Devido a um ambiente imprevisível, inconstante, e com uma alta variedade de ameaças, novos ataques começam a surgir a todo tempo no ambiente da nuvem. De acordo com o *Cloud Security Alliance* (HUBBARD; SUTTON et al., 2010) as principais ameaças para os modelos de serviço da nuvem são: Uso abusivo e ilegal da computação na nuvem, APIs e interfaces não seguras, usuários maliciosos, problemas com compartilhamento de tecnologia, perda ou vazamento de dados, sequestro de conta ou serviço, e perfil de risco desconhecido. Dentre as ameaças citadas, a mais preocupante se trata do uso abusivo e ilegal da computação na nuvem (HUBBARD; SUTTON et al., 2010), em que atacantes podem explorar as vulnerabilidades desse ambiente e utilizar seus recursos para executar um ataque.

Ataques de reconhecimento são responsáveis por escanear a nuvem em busca de vulnerabilidades que possam comprometer máquinas virtuais. Uma vez que as vulnerabilidades são identificadas, futuros ataques dentro das categorias de negação de serviço, remoto para local e usuário para super usuário podem ser lançados com o intuito de comprometer as VM. Ameaças dentro dessa categoria podem ser discretas, o que pode dificultar a sua detecção. De acordo com (LIPPMANN et al., 2000), um escaneamento é considerado discreto quando ele estabelece até no máximo dez conexões ou quando espera mais de 59 segundos entre sucessivas transmissões de rede.

Umas vez que estão comprometidos, os *hosts* de uma rede podem ser utilizados com o objetivo de executar a negação de serviço distribuído contra um determinado alvo. Ataques dentro dessa categoria podem afetar uma grande variedade de serviços, desde bancos *online* até *blogs* privados. Consequentemente, eles podem degradar a disponibilidade de um serviço até o ponto em que ele se torne inacessível (KALLIOLA et al., 2015). Atualmente, ameaças desse tipo estão aumentando em número e se tornando cada vez mais complexas. Elas estão focando os seus ataques para a camada de aplicação, contra aplicativos específicos de internet. Como consequência, esse tipo de ameaça consegue se esconder em tráfego normal de rede devido a uma baixa intensidade de tráfego requerida, dificultando a sua detecção (LIM et al., 2014).

Outras duas classes de ameaças que podem comprometer a nuvem explorando suas vulnerabilidades são remoto para local e usuário para super usuário. Sistemas de detecção dentro

da abordagem imunológica geralmente apresentam baixa eficácia para essas duas classes (JEYA; RAVICHANDRAN; RAVICHANDRAN, 2012). Isso se deve ao fato de que elas apresentam um número menor de conexões durante o tempo de ataque. Além disso, ataques desse tipo estão contidos no conteúdo de seus pacotes e, portanto, não consistem em uma sequência de padrões de tráfego, como no caso das classes DoS e *Probe* (JEYA; RAVICHANDRAN; RAVICHANDRAN, 2012).

Ataques que exploram vulnerabilidades englobam as quatro classes de ataques mencionadas anteriormente, pois exploram vulnerabilidades de uma rede para comprometer os sistemas presentes nela. Dentre os ataques sofridos constantemente no ambiente da nuvem, esse é considerado um dos ataques mais difíceis de detectar. Isso se deve ao fato de que usuários podem instalar aplicações vulneráveis a qualquer momento em suas máquinas virtuais, abrindo a possibilidade do surgimento de novas vulnerabilidades (CHUNG et al., 2013).

Além disso existem problemas associados a implantação de IDS tradicionais na nuvem, pois eles precisam constantemente de manutenção, não escalam bem com os requerimentos de segurança dos usuários, a manutenção deles é complicada e exigem tempo (AHMAD; IDRIS; KAMA, 2017). Outro fator se deve a análise de pacotes de rede, onde uma das técnicas utilizadas por eles é a de *Deep Packet Inspection*, na qual o conteúdo de pacotes inteiros é analisado. No entanto em ambientes de nuvens computacionais a taxa de transferência é muito alta, dificultando a análise para sistemas de detecção que analisam pacotes de rede (SEEBER; RODOSEK, 2015). Já em IDS baseados em anomalia é mais comum a análise do tráfego, onde a analise não ocorre a nível de pacote, facilitando a implantação desses sistemas. Uma vez que IDS dentro da abordagem AIS são baseados em anomalia, eles também possuem essa vantagem. Outras vantagens que sistemas de detecção baseados na abordagem imunológica possuem são a robustez relacionada a habilidade de detectar novos ataques, para que seja possível responder mais rapidamente em uma segunda reação ao mesmo ataque, e ausência de controle central, por se tratar de um sistema distribuído (LUTHER et al., 2007). Sendo assim, para esse trabalho foi selecionado um IDS dentro dessa abordagem, que utiliza os paradigmas de seleção negativa, seleção clonal, e rede imune.

A partir da implementação de um sistema de detecção de intrusão baseado em anomalia e dentro da abordagem imunológica, será possível melhorar a eficácia do IDS, assim como a seleção de contra-medidas, pois para a tomada de decisões serão utilizados dados relacionados ao ambiente no qual o sistema opera. Isso se deve ao uso de correlação de alertas por meio de grafo de ataque, uma vez que é levado em consideração a distribuição das vulnerabilidades presentes em uma rede.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver um modelo de segurança aplicado a nuvens computacionais e redes convencionais, usando a abordagem imunológica, capaz de combater ataques que exploram vulnerabilidades de um sistema, no qual o processo de seleção de contra-medidas é baseado nos conceitos de redes SDN e grafos de ataque.

1.2.2 Objetivos Específicos

- Implementar um sistema imunológico de detecção baseado em correlação de alertas onde são utilizados os principais paradigmas que se encontrem dentro da abordagem imunológica;
- Implementar um sistema de prevenção de intrusão em que a seleção e execução de contra-medidas se baseiam em conceitos de redes bayesianas e utilizam os recursos de grafos de ataque e redes SDN;
- Desenvolver uma abordagem de segurança na nuvem que possa integrar esses dois sistemas;
- Avaliar a proposta em relação a eficácia do IDS, visando aumentar a taxa de detecção, reduzir alarmes falsos, e selecionar contra-medidas mais apropriadas.

1.3 Justificativa

Atualmente para lidar com ataques que exploram vulnerabilidades, os administradores dos *datacenters* possuem controle sobre as máquinas virtuais, dessa forma as vulnerabilidades podem ser detectadas e corrigidas. No entanto, a correção de vulnerabilidades em sistemas em que usuários da nuvem possuem privilégios de controle de softwares instalados em suas VM, não são efetivas, uma vez que existe a possibilidade de instalação de softwares vulneráveis por parte dos usuários, afetando dessa forma a segurança (CHUNG et al., 2013). Além disso existe o risco de violação do *Service Level Agreement* (SLA) ou Acordo de Nível de Serviço.

Para prevenir que máquinas virtuais vulneráveis se tornem comprometidas na nuvem, foram desenvolvidas arquiteturas, ou modelos de segurança como os trabalhos de (CHUNG et al., 2013), e (MOUSSAID; TOUMANARI; AZHARI, 2017). Eles se baseiam em um modelo analítico de grafo de ataque para auxiliar no entendimento de ameaças, ajudando na seleção de contra-medidas apropriadas. A execução das contra-medidas selecionadas é realizada através da abordagem de redes SDN (*Software Defined Networking*), com o objetivo de melhorar a detecção e mitigar a consequência dos ataques. A melhora na detecção se deve ao uso de correlação

de alertas por meio de grafos de ataque. No entanto os trabalhos mencionados utilizam IDS baseados em assinatura, e não se preocupam em melhorar os algoritmos de detecção utilizados, mas sim em utilizar a abordagem de redes virtuais reconfiguráveis ou redes SDN, com o objetivo de detectar e combater tentativas de comprometer máquinas virtuais.

De forma a resolver os problemas enfrentados pelos artigos mencionados anteriormente (CHUNG et al., 2013) (MOUSSAID; TOUMANARI; AZHARI, 2017), nesse trabalho é proposto um modelo de segurança para a nuvens computacionais e redes convencionais. Nessa proposta, é utilizada uma técnica de correlação de alertas por meio de um grafo de ataque, em conjunto com um sistema de detecção baseado em anomalia e dentro da abordagem imunológica. A vantagem de se utilizar um IDS baseado em anomalia em relação a trabalhos que utilizaram essa técnica de correlação, se deve a automatização de respostas para novas ameaças, uma vez que esses sistemas não precisam de conhecimento prévio sobre determinada ameaça para que possa responder a ela. Dessa forma, se surgirem novos ataques relacionados a vulnerabilidades já existentes, ou relacionados a vulnerabilidades ainda não identificadas, o sistema de detecção ainda será capaz de combatê-los. Consequentemente, na presença de uma nova ameaça o sistema poderá ser notificado em tempo real, impedindo que contra-medidas importantes para proteção da rede deixem de ser executadas. Já uma vantagem existente em relação a utilização da técnica de correlação de alertas, se deve a diminuição de alarmes falsos, tendo em vista que esse se trata de um problema muito recorrente em IDS desse tipo.

Em relação a execução de contra-medidas, a abordagem proposta utiliza a tecnologia de redes SDN. Por possuir o *hardware* separado da lógica de controle, ela é capaz de reconfigurar a rede, através da execução de contra-medidas como redirecionamento ou isolamento do tráfego, filtro de pacotes, e bloqueio de portas TCP/UDP. E como consequência o modelo de rede baseado nessa tecnologia se torna mais flexível facilitando o combate a intrusões (KAUR et al., 2016).

1.4 Delimitação do Escopo da Pesquisa

O escopo da pesquisa se delimitou a pesquisar NIDS que se encontrem dentro da abordagem imunológica. Essa abordagem se enquadra na categoria de IDS baseados em anomalia. Esses sistemas possuem a habilidade de detectar novos tipos de ataques sem a necessidade de possuir conhecimento prévio sobre eles. Sendo assim, os que pertencem a abordagem imunológica tentam reproduzir funções do sistema imunológico humano, onde uma delas seria a capacidade de diferenciação entre *self* (Padrão normal) e *non-self* (Padrão anômalo). Os primeiros IDS baseados em AIS realizavam esse processo através da análise do tráfego da rede, principalmente a nível de pacote (LUTHER et al., 2007). Outra função está relacionada ao aprendizado na detecção de novos ataques, tornando possível uma resposta mais rápida em uma segunda reação a eles. Outro mecanismo encontrado seria o de casamento de padrões, no qual detectores casam com padrões anômalos. Existe também o processo de maturação, no qual os detectores aprendem

a diferenciar o *self* do *non-self*, geralmente representado pelo algoritmo de seleção negativa. Isso posto, no contexto de segurança, sistemas imunológicos artificias buscam reproduzir os principais atributos pertencentes ao HIS através da utilização de paradigmas como seleção negativa, seleção clonal, teoria do perigo e rede imune (LUTHER et al., 2007).

Outra delimitação da proposta está relacionada a utilização de redes SDN visando prover um ambiente mais seguro na nuvem. Essa tecnologia já foi utilizada tanto para auxiliar no processo de detecção de ameaças quanto no combate a elas. Um exemplo seria o trabalho de (SEEBER; RODOSEK, 2015), em que o procedimento de detecção consiste em um conjunto de passos. Cada passo no qual o IDS detecta uma intrusão, o tráfego dele é redirecionado através da abordagem SDN para outro IDS, e assim por diante, até a finalização dessa etapa. Já em relação ao combate, existe o trabalho de (XING et al., 2014), no qual a rede é reconfigurada através de alertas gerados pelo IDS Snort. Essa reconfiguração pode ocorrer por meio do redirecionamento ou isolamento do tráfego, pelo filtro de pacotes, pelo bloqueio de portas TCP/UDP, ou por ajustes na qualidade de serviço (QoS - *Quality of Service*). QoS se refere a um sistema de medida do desempenho de um determinado serviço, em que no contexto desse trabalho seriam serviços fornecidos pela nuvem. Sendo assim, conforme apresentado pelos trabalhos citados, a abordagem de redes SDN oferece muitos benefícios pois separa o *hardware* da lógica de controle, e como consequência é possível obter um modelo de rede mais flexível e seguro (KAUR et al., 2016).

A abordagem de segurança proposta nesse trabalho irá monitorar o tráfego gerado entres as máquinas virtuais, onde no caso de identificação de algum ataque ou comprometimento de alguma VM, serão executadas contra-medidas de segurança baseadas nos conceitos de redes SDN. Uma vez que o modelo proposto visa proteger a infraestrutura da nuvem, ou seja as máquinas virtuais, ele é focado na segurança em IaaS. Esse modelo de serviço oferece recursos de *hardware* para seus usuários, através de *softwares* de virtualização, conhecidos como hypervisors, cuja função está relacionada a criação e gerenciamento de máquinas virtuais. Através da virtualização é possível aproveitar melhor os recursos de *hardware* oferecidos pela nuvem, e como consequência é possível atender a uma demanda maior (SABAHI, 2012). Isso posto, a principal vantagem oferecida por esse modelo está relacionada a economia que o cliente irá obter uma vez que a infraestrutura de processamento, armazenamento, e rede, não precisará ser adquirida ou mantida.

Sendo assim, esse trabalho se delimitou a desenvolver uma abordagem na qual a etapa de detecção se encontra dentro da abordagem imunológica. Seguido por uma técnica de correlação de alertas baseada em grafos de ataque para redução de alarmes falsos. Enquanto a etapa de prevenção, ou seja, a execução de contra medidas se delimita à utilização da tecnologia de redes SDN, cuja abordagem se foca em proteger a infraestrutura da nuvem, visando combater ataques que exploram vulnerabilidades de máquinas virtuais, comprometendo-as pela negação de serviço distribuído (DDoS - *Distributed Denial of Service*) em larga escala.

1.5 Estrutura do Trabalho

A estrutura do trabalho é composta por oito capítulos, sendo a introdução o primeiro deles. Esse primeiro capítulo introduz o assunto e trata da problemática e hipótese, objetivos, justificativa e delimitação do escopo referentes à pesquisa sendo realizada. No capítulo 2 são apresentadas as principais abordagens, técnicas, e conceitos utilizados dentro do tema pesquisado. Em seguida, no capítulo 3 é discutida a metodologia de pesquisa. No capítulo 4 os trabalhos relacionados são discutidos e comparados entre si e com a abordagem proposta. No capítulo 5 é apresentada a proposta de dissertação. No capítulo 6 os cenários experimentais da pesquisa são apresentados. O capítulo 7 apresenta os resultados experimentais. Por último, no capítulo 8 é apresentada as conclusões seguidas de discussões sobre trabalhos futuros.

2 Referencial Teórico

Nesta seção será abordado o tema de computação nas nuvens, pois trata-se do ambiente no qual a abordagem de segurança proposta irá operar. Sendo assim, é importante entender sobre os serviços e modelos de implantação disponibilizados por esse ambiente, assim como também os ataques aos quais eles são submetidos. Também é necessário entender sistemas de detecção e de prevenção, assim como conhecer as diferencias entre os dois, pois no modelo de segurança proposto esses sistemas serão implantados. Além disso, é importante compreender acerca de sistemas imunológicos artificiais e os principais algoritmos utilizados por eles na área de segurança, por se tratar do paradigma escolhido no processo de detecção de ameaças no modelo de segurança proposto. Em seguida é importante entender sobre o processo de correlação de alertas, onde cada alerta gerado pelo sistema de detecção será mapeado para um grafo de ataque. Por último, é preciso entender sobre o processo de seleção de contra-medidas, no qual é baseado em um algoritmo que utiliza a a abordagem de redes Bayesianas, onde a execução de cada uma delas ocorre por meio da tecnologia de redes SDN, que se trata da ultima seção discutida nesse capítulo.

2.1 Computação em Nuvens

Na computação em nuvens, usuários possuem acesso a serviços sem o conhecimento de onde eles estão hospedados, ou de como eles são fornecidos. Nesse ambiente, eles são capazes de acessar a aplicativos de qualquer parte do mundo, sem a preocupação da infraestrutura por trás desse sistema. Essa infraestrutura consiste de datacenters, ou conjunto de servidores interligados, que são monitorados ou mantidos por fornecedores de conteúdo (BUYYA et al., 2009). Esse ambiente fornece três tipos de serviços que são conhecidos como software como serviço (SaaS - *Software as a Service*), plataforma como serviço (PaaS - *Platform as a Service*), e infraestrutura como serviço (IaaS - *Infrastructure as a Service*). Consumidores, como empresas, são atraídos pela redução ou eliminação de custos relacionados à sua implantação e manutenção. No entanto, como aplicações na nuvem apresentam-se essenciais para os negócios de uma empresa, existe a necessidade de estabelecer acordo legal entre as partes*Service Level Agreements* (SLAs). Esse acordo possui o objetivo de certificar que os consumidores possuam garantias dos fornecedores de serviços (BUYYA et al., 2009).

No modelo SaaS, os clientes possuem acesso remoto a aplicativos que se encontram instalados na nuvem, mas a possessão do *software* é separada de sua propriedade. Fornecendo dessa forma a funcionalidade de um *software* como um conjunto de serviços distribuídos que podem ser adquiridos e configurados dentro de seu tempo de uso. Esse modelo visa solucionar problemas relacionados aos custos de sua implantação e manutenção. (TURNER; BUDGEN;

BRERETON, 2003). No entanto, esse modelo apresenta preocupações em relação a forma como os dados são armazenados e protegidos, uma vez que eles se encontram armazenados fora dos limites das empresas, resultando em desinteresse por parte delas em adotar esse tipo de serviço (SUBASHINI; KAVITHA, 2011). Outro fator que contribui para esse desinteresse se deve à falta de garantia de ausência de problemas no *software*, onde caso algum *bug* seja identificado, o contratante do serviço não poderá soluciona-lo, pois não irá dispor de informações suficientes devido a questões de privacidade (GOLD et al., 2004).

O serviço PaaS, por sua vez, fornece acesso a um ambiente de desenvolvimento remoto, possibilitando o desenvolvimento de projetos de software sem a necessidade de um *framework* instalado localmente. Nesse serviço o fornecedor escolhe como as aplicações serão desenvolvidas a partir da determinação de um conjunto de ferramentas de desenvolvimento, como APIs (*application programming interface*), IDEs (*integrated development environment*), assim como também do sistema operacional sob o qual as aplicações serão desenvolvidas e a linguagem de programação utilizada (LAWTON, 2008). As vantagens desse modelo se devem ao aumento de produtividade, diminuição no tempo de conclusão de projetos, e redução nos custos (LAWTON, 2008). No entanto, uma desvantagem se deve a abertura que esse serviço fornece em relação ao desenvolvimento de aplicativos utilizando plataformas remotas, uma vez que pode facilitar usuários mal-intencionados a desenvolver aplicativos maliciosos que comprometam a infraestrutura da nuvem. E por isso é preciso implementar mecanismos de segurança confiáveis que se encontrem abaixo do nível dessas aplicações, como sistemas de prevenção de redes, por exemplo (SUBASHINI; KAVITHA, 2011). Outro problema está relacionado a possibilidade de um sistema PaaS parar de funcionar e não suportar acesso *offline* as aplicações, ou também pode ocorrer falência do provedor PaaS, resultando em uma situação crítica para a empresa contratante. Além disso existe a questão dos usuários se encontrarem presos a uma única plataforma de desenvolvimento, dificultando a migração de suas aplicações para outra plataforma. Um exemplo dessa necessidade poderia surgir em decorrência de um aumento nos preços oferecidos pelo provedor. Por último, também existe a questão de que nem sempre uma plataforma de desenvolvimento provê todas as funcionalidades necessárias para o desenvolvimento de uma aplicação (LAWTON, 2008).

Por último, existe o serviço de infraestrutura ou IaaS, no qual a infraestrutura da nuvem é virtualizada, onde por meio desse processo a sua infraestrutura pode ser utilizada por meio de máquinas virtuais (DESVENDANDO. . . , 2017). A técnica de virtualização é utilizada por *softwares* denominados *Hypervisors*. Por meio deles é possível que múltiplos sistemas operacionais, ou hóspedes, executem concorrentemente em um único computador. O *software* de virtualização ou *hypervisor* fornece uma plataforma virtual para monitorar a execução desses hóspedes. Múltiplas instâncias de uma variedade de sistemas operacionais compartilham os recursos do *hardware* virtualizado no qual se encontram hospedados (SABAHI, 2012). A partir do processo de virtualização é possível aproveitar melhor os recursos oferecidos pelo *hardware*, uma vez que esses recursos serão capazes de atender a uma demanda maior. Sendo assim, esse modelo de serviço

possui as vantagens de redução de custos, uma vez que não será necessário implantar ou manter uma infraestrutura que forneça poder de processamento, armazenamento, e uma estrutura de rede, uma vez que esses recursos podem ser oferecidos pelo serviço de infraestrutura (BHARDWAJ; JAIN; JAIN, 2010). No entanto uma desvantagem se deve a casos em que vulnerabilidades no *software* de virtualização podem ser exploradas por usuários mal-intencionados, comprometendo todos os modelos de serviço da nuvem, uma vez que o modelo IaaS é base para todos os outros tipos de serviços fornecidos (SUBASHINI; KAVITHA, 2011).

Além dos tipos de serviços que a nuvem fornece ainda existem os modelos de implantação que podem ser de nuvem pública, nuvem privada, nuvem de comunidade, e nuvem híbrida. Esses modelos estão relacionados ao acesso e disponibilidade providos por esse ambiente, e a restrição ou abertura do acesso depende de fatores como processo de negócio, tipo de informação, e nível de visão (SOUSA; MOREIRA; MACHADO, 2009). Pois existem empresas que restringem seus recursos a apenas seus funcionários, ou existem outras que liberam apenas partes de seus recursos. Sendo assim existe a necessidade de categorizar sistemas de nuvens computacionais de acordo com o seu nível de acesso. Na nuvem pública, por exemplo, os servidores são compartilhados e acessados pela internet, onde qualquer cliente possui acesso (DESVENDANDO. . . , 2017). Nesse modelo, geralmente os três tipos de serviço são oferecidos aos clientes a um determinado custo. Nele, não podem ser aplicadas restrições de acesso quanto ao gerenciamento de redes, e menos ainda, utilizar técnicas para autenticação e autorização (SOUSA; MOREIRA; MACHADO, 2009). Enquanto a nuvem privada é implantada por uma empresa que exige que a sua utilização se dê apenas pelos seus funcionários, no entanto essa estrutura implantada pode ser local ou remota, e administrada pela própria empresa ou por terceiros. Sendo assim nesse modelo são empregadas políticas de utilização dos seus serviços. As técnicas utilizadas para prover tais características podem ser a nível de gerenciamento de redes, de configuração dos provedores de serviços, ou de utilização de tecnologias de autenticação e autorização (SOUSA; MOREIRA; MACHADO, 2009). Existe também a nuvem de comunidade, onde esse ambiente é compartilhado com diversas empresas que partilham interesses relacionados a missão, requisitos de segurança, política e considerações de flexibilidade (SOUSA; MOREIRA; MACHADO, 2009). Por último a nuvem híbrida possui uma combinação de uma ou mais estruturas em nuvem, que podem ser privada, de comunidade ou pública, compondo assim uma entidade única, que podem se encontrar ligadas por uma tecnologia padronizada ou proprietária que permite a portabilidade de dados e aplicações (SOUSA; MOREIRA; MACHADO, 2009).

Os modelos de serviço e implantação na nuvem podem ser visualizados de acordo com as figuras 1 e 2.

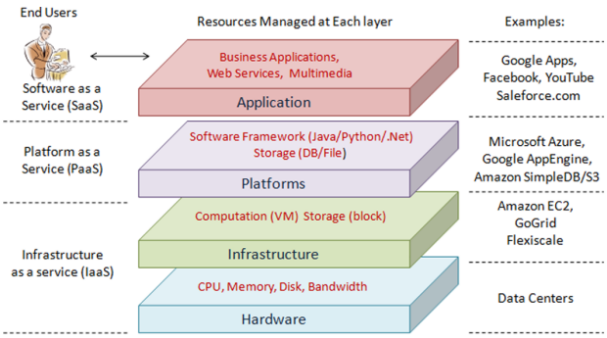


Figura 1 – Arquitetura em Camadas do Modelo de Serviços (ZHANG; CHENG; BOUTABA, 2010)

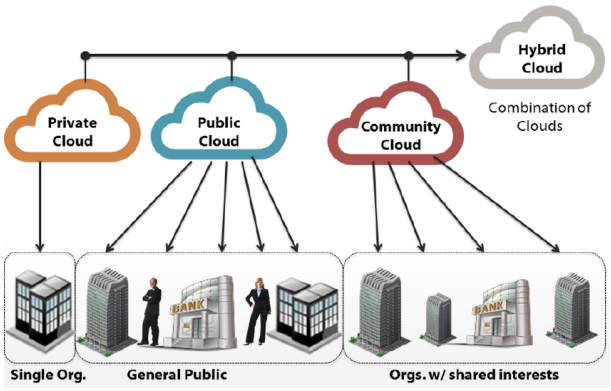


Figura 2 – Modelos de Implantação (CARPINTERIA. . . , 2017)

2.2 Ataques

Essa seção explica sobre ataques que exploram vulnerabilidades que podem se enquadrar em quatro categorias: negação de serviço, ataques de reconhecimento, remoto para local, e usuário para super usuário. Entre as quatro categorias, a de negação de serviço envolve ataques de negação de serviço distribuído (DDoS - *Distributed Denial of Service*) que são consequência de ataques de reconhecimento, cuja função é de sondar a infraestrutura da nuvem em busca de vulnerabilidades. Dessa forma, é possível explorar as vulnerabilidades detectadas e como consequência comprometer as máquinas virtuais. Sendo assim, ataques DDoS geralmente envolvem estágios iniciais relacionados a exploração da infraestrutura, como exploração em múltiplas etapas, escaneamento de baixa frequência de vulnerabilidades, comprometimento das máquinas virtuais, e por fim ataques DDoS através das VM comprometidas (CHUNG et al., 2013). Já a categoria de remoto para local envolve ameaças que permitem o acesso não autorizado de uma máquina remota para outra local. A categoria de usuário para super usuário, também conhecida como ataques que escalam privilégios, envolvem ameaças que permitem acesso não autorizado de super usuário para uma máquina local (AMBEDKAR; BABU, 2015).

As ameaças abordadas nessa seção estão relacionadas as quatro categorias de ataques que sistemas de detecção baseados em anomalia buscam combater, assim como a ataques que buscam

explorar vulnerabilidades de sistemas de nuvens computacionais. Essas vulnerabilidades podem estar associadas a infraestrutura da nuvem, a aplicativos *web*, a protocolos de sincronização, a *tokens* de identificação, ou a processos de validação de assinaturas de servidores *web*. Dessa forma, através do modelo de segurança proposto será possível identificar alterações no tráfego causadas pela exploração das vulnerabilidades no ambiente da nuvem. No caso de identificação de alguma alteração, contra-medidas associadas a tecnologia de redes SDN poderão ser executadas. Onde por exemplo, no comprometimento de alguma máquina virtual, ela poderia ser isolada da rede, dificultando o comprometimento de outras VM, mitigando dessa forma danos causados em um possível cenário de ataque. Outra utilidade para essa tecnologia seria no auxílio ao processo de detecção, onde o fluxo de tráfego de uma máquina suspeita poderia ser redirecionado com o propósito de se aplicar técnicas como o *Deep Packet Inspection* na análise dos pacotes da rede.

2.2.1 Ataques de Reconhecimento

Ataques de reconhecimento envolvem estágios iniciais, como sondagem de uma rede para descobrir quais *hosts* estão ativos (Varredura de Ip) (AMBEDKAR; BABU, 2015). Em seguida, o invasor escaneia diversos *hosts* para determinar se uma porta específica está aberta (Varredura de porta). Alternativamente, um varredura pode ser realizada em um *host* específico para determinar qual das portas estão abertas (Escaneamento de porta). Finalmente, o atacante analisa quais máquinas são vulneráveis para cada serviço executando em uma porta específica, a fim de lançar um ataque e comprometer o hospedeiro. Dessa forma o processo de sondagem pode servir como passo inicial para outras categorias de ataque, como negação de serviço, remoto para local, e usuário para super usuário.

Ataques desse tipo podem apresentar dificuldades em sua detecção devido a sua furtividade. De acordo com (LIPPMANN et al., 2000), uma sondagem é considerada furtiva se ela emite dez ou menos conexões ou pacotes, ou se ela espera mais de 59 segundos entre transmissões sucessivas na rede.

2.2.2 Negação de Serviço

Ataques de negação de serviço, cujo objetivo se trata de impedir que usuários legítimos acessem recursos da rede, são conhecidos desde a década de 80 (ZARGAR; JOSHI; TIPPER, 2013). Em 1999 de acordo com o CIAC (*Computer Incident Advisory Capability*), foi reportado o primeiro ataque de negação de serviço distribuído, e a partir dessa data a maior parte desses ataques passaram a ser distribuídos (ZARGAR; JOSHI; TIPPER, 2013). Atualmente eles podem ocorrer através de pacotes malformados que são enviados para a vítima, impedindo o funcionamento correto de algum protocolo ou aplicação que execute no seu sistema. Através do esgotamento da largura de banda prejudica a conexão do usuário ou exaure os recursos da nuvem, provocando a negação de serviço (ZARGAR; JOSHI; TIPPER, 2013).

Atualmente esses ataques podem ocorrer a partir de uma rede de máquinas remotamente controladas que enviam uma grande quantidade de solicitações de serviço para o seu alvo, provocando o seu colapso. Através da utilização dos recursos de outras máquinas comprometidas é possível lançar um ataque DDoS em larga escala, em que a origem do ataque se torna difícil de identificar devido a falsificação dos endereços IP das máquinas comprometidas (ZARGAR; JOSHI; TIPPER, 2013). Outra tendência está associada ao aumentando em número e complexidade para ameaças desse tipo. Por essa razão, elas estão impactando negativamente uma variada quantidade de aplicações da Internet (BURAGOHAIN; MEDHI, 2016). Isso afeta negativamente a eficácia da detecção, já que atualmente eles focam uma aplicação específica na camada de aplicação. Como resultado, tais ameaças direcionadas podem facilmente se esconder em tráfego normal devido à baixa intensidade de ataque requerida (LIM et al., 2014). Além disso, DDoS modernos exploram um grande número de *hosts* para focarem em um alvo específico. Dessa forma eles emitem solicitações de serviço com aparência legítima para o servidor alvo, resultando em tráfego anômalo similar à tráfego normal (LIM et al., 2014).

2.2.3 Ataques Remoto para Local e Usuário para Super Usuário

Ataques R2L consistem em ameaças que permitem o acesso não autorizado de uma máquina remota para outra local (AMBEDKAR; BABU, 2015). Como consequência, o invasor passa a possuir acesso a máquina da vítima com privilégios de um usuário local. Enquanto a categoria de U2R, também conhecida como ataques que escalam privilégios, envolvem ameaças que permitem acesso não autorizado de super usuário para uma máquina local (AMBEDKAR; BABU, 2015).

Ambas as categorias constituem um grupo de ameaças que apresentam dificuldades em sua detecção devido ao fato de que possuem um número menor de conexões durante o tempo de ataque. Além disso, ataques desse tipo estão contidos no conteúdo de seus pacotes e, portanto, não consistem em uma sequência de padrões de tráfego, como no caso das classes DoS, e *Probe* (JEYA; RAVICHANDRAN; RAVICHANDRAN, 2012). A seguir serão apresentados ataques importantes que pertencem a esse grupo.

2.2.4 Buffer Overflow

De acordo com (Imperva, 2018), ataques de *buffer overflow* envolvem o envio de fluxos de entrada excessivamente longos para o servidor alvo, fazendo com que ele transborde partes da memória e que trave o sistema ou execute o código arbitrário do invasor como se fosse parte do código do servidor. O resultado é um comprometimento total do servidor ou negação de serviço.

2.2.5 Malware Injection

Nesse tipo de ataque os *hackers* exploram vulnerabilidades de aplicações *web* visando introduzir código malicioso nelas, comprometendo dessa forma a execução desses aplicativos. Sendo assim, da mesma forma que aplicações *web* são suscetíveis a ataques de *Malware Injection*, sistemas na nuvem também são. Consequentemente, é possível desenvolver aplicativos, programas, e máquinas virtuais maliciosos com o objetivo de comprometer os serviços fornecidos nesse ambiente, ou seja, IaaS, PaaS, e SaaS. Uma vez que o código é introduzido nesses serviços, ele passa a executar como uma instância válida na nuvem, possibilitando a espionagem, manipulação e roubo dos dados (CHOU, 2013).

Ainda de acordo com (CHOU, 2013), dentro da categoria de ataques *Malware Injection* as formas mais comuns de ataque são SQL *injection*, e *cross-site scripting*. O primeiro consiste em comprometer servidores SQL que executam aplicações de banco de dados vulneráveis. Isso se deve a introdução de código malicioso nesses servidores para burlar o *login* e ganhar acesso não autorizado as bases de dados. Por meio desse acesso os *hackers* podem manipular os dados dessas bases, obter informações confidenciais, ou executar sistemas de comando remotamente. Uma vítima desse tipo de ataque se trata do Play Station da Sony, em que foram introduzidos código malicioso em 209 de suas páginas visando promover jogos como "God of War"e "SingStar Pop"(VISITORS. . . , 2017). Já em ataques *cross-site scripting* são injetados *scripts* maliciosos, como JavaScript, HTML, e Flash, em uma página *web vulnerável* para que ele execute no navegador da vítima, possibilitando atividades ilegais para acessar a conta da vítima ou para induzi-la a clicar em *links* maliciosos (CHOU, 2013).

2.3 Sistemas de Detecção e Prevenção de Intrusão

No decorrer das décadas começaram a surgir inúmeros problemas de segurança relacionados a redes e sistemas computacionais (LIAO et al., 2013), e devido a isso novas técnicas de segurança começaram a aparecer, entre elas podemos destacar os sistemas de detecção, e de prevenção. Segundo (KEMMERER; VIGNA, 2002), sistemas de detecção de intrusão são complementares aos *firewalls*. Enquanto o segundo oferece políticas de filtro de pacotes, ele ainda pode falhar ou podem esquecer de atualizá-lo. Por isso existem os IDS, quando uma falha ocorre eles são capazes de detectar uma intrusão. Eles conseguem detectar evidencias de intrusão, quer elas se encontrem em progresso ou depois de ocorridas. Essa evidência pode ser referida algumas vezes como manifestação de ataque. Se não há essa manifestação, se existe poucas provas dessa manifestação, ou se as provas não são confiáveis, esses sistemas podem falhar (KEMMERER; VIGNA, 2002). Já um sistema de prevenção de intrusão se trata de qualquer dispositivo, *hardware* ou *software*, que possui a habilidade de detectar e prevenir que um ataque seja bem sucedido (IERACE; URRUTIA; BASSETT, 2005). Ele irá executar contra-medidas para minimizar ou impedir que danos sejam causados em um cenário de intrusão. Esses sistemas

são importantes porque podem impedir ataques em tempo real, além de proteger recursos valiosos de empresas. Por isso cada vez mais companhias vem adotando sistemas de prevenção (IERACE; URRUTIA; BASSETT, 2005).

Sistemas de detecção podem se dividir entre os baseados em anomalia e os baseados em assinatura. Na abordagem baseada em assinatura, um padrão ou *string*, também conhecido como assinatura, é utilizado para representar o conhecimento acumulado acerca de ataques ou vulnerabilidades do sistema, que serão comparados com eventos capturados, possibilitando detectar uma intrusão (LIAO et al., 2013). Dessa forma, assim que um novo ataque surge o seu padrão é estudado e uma assinatura é gerada para o seu reconhecimento. Ela pode ser representada por um trecho de caracteres dentro de um código malicioso, pelos recursos alvos de uma intrusão, ou por um padrão de ataque (KABIRI; GHORBANI, 2005). Consequentemente, depois do surgimento da nova intrusão, sua assinatura é estudada por especialistas em segurança para atualizar o IDS, tornando-o capaz de reconhecê-la. Esse método é bem eficiente para ataques conhecidos por possuir uma quantidade baixa de alarmes falso positivos, no qual o alarme é gerado na ausência de intrusão (KABIRI; GHORBANI, 2005). No entanto, a sua principal desvantagem se deve a inabilidade de detectar novos ataques pois não existe conhecimento prévio sobre eles. Uma pequena alteração na assinatura do ataque já é suficiente para que esses sistemas não consigam detecta-lo. Já IDS baseados em anomalia são capazes de detectar ataques novos, no entanto possuem a desvantagem de possuir uma quantidade elevada de alarmes falso positivos.

De acordo com (LIAO et al., 2013), sistemas de detecção baseados em anomalia utilizam um perfil para representar comportamento normal, onde não há presença de intrusão, e uma anomalia é representada por um desvio do comportamento conhecido de um sistema, onde o monitoramento ocorre através do tráfego de rede, máquinas ou usuários. Já (GARCIA-TEODORO et al., 2009) explica que NIDS baseados em anomalia se dividem em três etapas diferentes, que são: parametrização, estágio de treinamento, e estágio de detecção. O primeiro se preocupa em observar as instâncias do sistema alvo para que ele possa ser representado de uma forma pré-estabelecida. Enquanto no estágio de treinamento o comportamento normal e anormal do sistema é definido e um modelo de representação correspondente é desenvolvido, esse processo pode ser realizado automaticamente ou manualmente. Já na última etapa, uma vez que o modelo para o sistema está disponível, ele é comparado com o tráfego parametrizado, onde no caso de haver um desvio maior do que um determinado limiar definido, um alarme será gerado. A figura 3 apresenta as etapas descritas.

Ainda de acordo com (GARCIA-TEODORO et al., 2009) NIDS dentro da abordagem de anomalia podem ser categorizados de acordo com as técnicas utilizadas, que podem ser baseadas em estatística, conhecimento, e aprendizado de máquina. O primeiro se preocupa em capturar o tráfego de rede com o objetivo de elaborar um perfil representando seu comportamento. Esse perfil pode ser representado por métricas como a taxa do tráfego, número de pacotes para cada

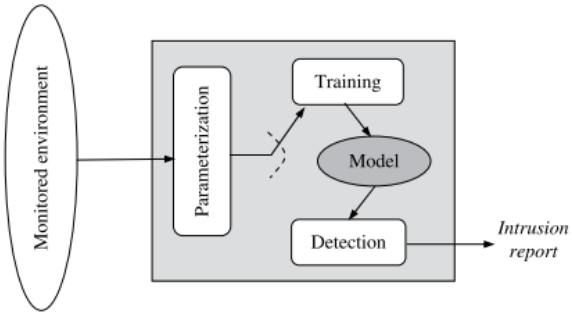


Figura 3 – Etapas de NIDS baseados em Anomalia (GARCIA-TEODORO et al., 2009)

protocolo, taxa de conexões, e número de diferentes endereços IP. Dessa forma, dois conjuntos de dados são considerados para o processo de detecção, onde um corresponde ao perfil do tráfego sendo observado no momento, enquanto o outro corresponde ao perfil de treinamento, no qual é representado o tráfego normal da rede. Sendo assim, na medida que os eventos de tráfego ocorrem, o perfil observado é determinado e comparado com o perfil de treinamento. Por meio dessa comparação é calculado o nível de anormalidade de um determinado evento, e no caso desse valor exceder o limiar estabelecido, será gerado um alarme indicando a ocorrência de uma anomalia. Já o baseado em conhecimento classificam os dados de acordo com um conjunto de regras. Primeiramente, diferentes atributos e classes são identificados a partir dos dados de treinamento. Em seguida, um conjunto de regras, parâmetros, ou procedimentos são desenvolvidos, onde logo após, os dados serão classificados de acordo com as especificações elaboradas. A elaboração das especificações pode ser construída manualmente ou automaticamente, onde um exemplo de construção automática seria através de alguma ferramenta de máquina de estado finito, no qual uma sequência de estados e transições seriam utilizados para modelar protocolos de rede (ESTEVEZ-TAPIADOR; GARCIA-TEODORO; DIAZ-VERDEJO, 2003). No entanto para NIDS desse tipo a principal desvantagem se deve ao alto consumo de tempo e dificuldade de elaboração dessas especificações (SEKAR et al., 2002). Enquanto os baseados em aprendizado de máquina se concentram em desenvolver um modelo comportamental no qual é possibilitada a categorização dos padrões analisados. Onde uma característica singular dos sistemas desse tipo se deve a necessidade de dados rotulados, com a função de treinar o modelo desenvolvido. A principal diferença dessa técnica com a baseada em estatística é que nessa o modelo desenvolvido é capaz de melhorar sua eficácia através de resultados anteriores. Exemplos de técnicas de aprendizagem que podem ser utilizadas por NIDS dentro da abordagem de anomalia são: redes neurais, redes bayesianas, e algoritmo genético (GARCIA-TEODORO et al., 2009). A tabela 1 realiza um comparativo entre as abordagens de anomalia e assinatura.

Sistemas de prevenção de intrusão (IPS - *Intrusion Prevention System*) combinam a função de detecção dos IDS com a função de prevenção em que contra-medidas são executadas

para conter uma intrusão. Esses sistemas podem responder aos ataques alterando o conteúdo deles, onde porções do ataque poderiam ser removidas tornando-o benigno, ou através da alteração do ambiente monitorado, modificando a configuração de dispositivos de segurança. Por exemplo, um dispositivo de rede poderia ser reconfigurado para bloquear o acesso de um atacante ou de uma vítima, ou o *firewall* de uma máquina poderia ser alterado para impedir possíveis intrusões (PATEL et al., 2013). Porém, um desvantagem de IPS está associada a geração de alarmes falso positivos, pois através deles o sistema pode executar uma contra-medida onde por meio dela, atividades da rede poderiam ser bloqueadas causando a negação de serviço para um usuário válido (IERACE; URRUTIA; BASSETT, 2005). De acordo com (PATEL et al., 2013), outros problemas desses sistemas que estão relacionados diretamente a etapa de prevenção no ambiente da nuvem, são a remoção ou adição dinâmica de VM, onde como consequência os seus requerimentos de segurança tendem a ser variados, podendo prejudicar no desempenho da prevenção. Além do alto número de administradores de segurança nesse ambiente, diminuindo o tempo de resposta dos IPS, devido a intervenção humana.

Tabela 1 – Comparativo Entre Abordagens de Anomalia e Assinatura

Abordagens	Vantagens	Desvantagens
Assinatura	1. Método simples e efetivo para a detecção de ataques conhecidos; 2. Análise contextual detalhada.	1. Ineficiente para detectar ataques desconhecidos, ataques de evasão, ou variação de ataques conhecidos; 2. Pouco entendimento de estados e protocolos; 3. Difícil manter assinaturas/padrões atualizados; 4. Consome muito tempo para manter a base de dados atualizada.
Anomalia	1. Eficiente em detectar novas vulnerabilidades; 2. Menos dependente de sistema operacional; 3. Facilita detecções de abuso de privilégio.	1. Baixa precisão do perfil normal devido a eventos que variam constantemente; 2. Indisponível durante a reconstrução do perfil normal; 3. Dificuldade de gerar alertas no tempo certo.

2.4 Sistemas Imunológicos Artificiais

Modelo baseado no sistema imunológico humano, inspirado em seus princípios de aprendizado e memória para aplicação na solução de problemas. Trata-se de uma área relativamente nova que surgiu a partir de trabalhos realizados por diversos imunologistas teóricos (JERNE, 1974) (PERELSON, 1989) (BERSINI; VARELA, 1991). O interesse dos pesquisadores não está na modelagem do sistema imune mas na compreensão dos mecanismos por trás dele que possam

ser utilizados como inspiração no desenvolvimento de ferramentas computacionais visando solucionar problemas específicos. Sendo assim, essa abordagem possui diversos atributos que podem despertar o interesse dos pesquisadores.

De acordo com (TIMMIS et al., 2004) existem diversos fatores que contribuem para que a abordagem imunológica seja de interesse para a computação. Entre eles, podemos citar primeiramente a habilidade de reconhecimento. O sistema imune é capaz de reconhecer diversos padrões diferentes. Além disso, através dele é possível distinguir entre células defeituosas, pertencentes ao próprio sistema, de células prejudiciais que não pertencem ao sistema. Extração de atributos, no qual são extraídos atributos de partículas ou moléculas, conhecidas como antígenos que são capazes de deflagrar a produção de anticorpos específicos. Diversidade, referente à habilidade do sistema imune de possuir uma ampla cobertura no reconhecimento de antígenos, pois ele produz uma grande diversidade de anticorpos. Aprendizagem, pois através do processo de afinidade por maturação ele consegue se tornar cada vez melhor no reconhecimento de padrões. Memória, uma vez que uma segunda resposta a algum antígeno será mais rápida, já que foram armazenadas informações sobre ele, onde esse processo é conhecido como maturação de resposta imune. Detecção distribuída, já que não existe um ponto de controle central, e dessa forma cada célula imune pode responder separadamente a um determinado antígeno. Auto-regulação, pois a população do sistema imune é controlada por interações locais, que pode variar com a presença ou ausência de alguma doença. Meta-dinâmica, já que novas células ou moléculas estão sempre sendo criadas pelo sistema imune com o intuito de substituir aquelas que são muito velhas ou que não são úteis. Por último existe a rede imune que se trata de um processo de rede interna de comunicação na qual células e moléculas do sistema imune estão reagindo entre elas.

Atualmente essa abordagem vem sendo cada vez mais empregada em diversas áreas, entre elas podemos citar, por exemplo, problemas de classificação ou de detecção de anomalias. Na área de detecção de anomalias podemos elencar sistemas de detecção de intrusão, relacionado ao tema desse trabalho, ou detecção de fraudes de cartão de crédito. Este último (HALVAIEE; AKBARI, 2014) introduziu um novo método baseado em AIS denominado *AIS-based Fraud Detection Model* (AFDM) com a função de diminuir o tempo de treinamento. Além disso eles melhoram um algoritmo imune apresentado em (WATKINS; TIMMIS; BOGGESS, 2004) para atingir uma maior precisão. Já na área de classificação podemos citar os trabalhos de (COOKE; HUNT, 1995) (HUNT; COOKE, 1996) no qual os autores procuram desenvolver um mecanismo de aprendizagem de máquina supervisionado visando classificar sequências de DNA através da criação de *strings* de anticorpos. Outro exemplo aborda a classificação de imagens de sensoriamento remoto (YAN; ZHONG, 2008), onde as mesmas fornecem informações de utilização ou de cobertura de uma larga área geográfica.

De acordo com os estudos realizados nesse trabalho, foi possível perceber que apesar dessa área possuir aplicabilidade em diversos campos, ainda não foi encontrado um *framework*

geral que possibilite o desenvolvimento de sistemas imunológicos artificias e que explique como todos os paradigmas de AIS utilizados poderiam ser combinados em uma única arquitetura. Diferente de outras áreas de inteligência computacional como redes neurais, computação evolutiva ou sistemas *fuzzy*, onde existe um conjunto de componentes ou mecanismos bem descritos que auxiliam no desenvolvimento de técnicas ou algoritmos (TIMMIS et al., 2004).

2.4.1 Principais Abordagens de IDS baseados em AIS

A primeira questão que deve ser tratada na implementação de um IDS baseado em AIS é o de representação do domínio do problema, onde o mesmo será representado por *self* (Comportamento normal do sistema) ou *non-self* (Comportamento anômalo do sistema). Em seguida serão definidas as regras de casamento de padrões, com o objetivo de distinguir esses dois grupos. Um dos primeiros e principais algoritmos utilizados para esse problema é o de Seleção-Negativa (NSA - *Negative Selection Algorithm*). Na abordagem original desse algoritmo (FORREST et al., 1994) é utilizado codificação binária, para simular a relação anticorpo/antígeno, onde os anticorpos seriam os detectores responsáveis por detectar comportamento anômalo (antígenos). Nesse modelo os detectores são gerados aleatoriamente, enquanto o *self* é constantemente monitorado, uma vez que em um ambiente mutável ele estará em constate mudanças, com a função de eliminar qualquer detector que case com um dos elementos do *self*. Por último os detectores selecionados, monitoram constantemente o ambiente em busca de algum novo padrão anômalo que represente alguma ameaça. A figura 4 apresenta as ações do algoritmo de seleção negativa.

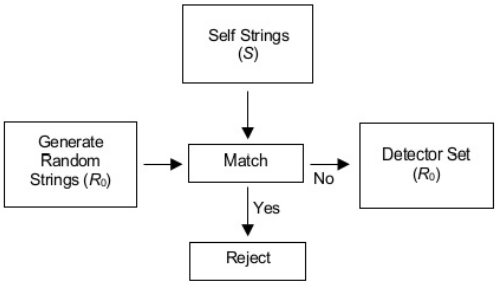


Figura 4 – Algoritmo de Seleção Negativa (FORREST et al., 1994)

A segunda questão a ser tratada é referente à geração dos detectores, com o objetivo de selecionar aqueles que não casaram com nenhum elemento do *self*. Segundo a abordagem do NSA descrita por (FORREST et al., 1994), eles são gerados aleatoriamente. No entanto existem técnicas que podem ser utilizadas nessa fase, que resultam em um melhor desempenho para esses sistemas. De acordo com o trabalho de (D'HAESELEER; FORREST; HELMAN, 1996), duas das técnicas utilizadas são a de algoritmo linear e algoritmo guloso. Na primeira, padrões de *strings* que não irão se tornar detectores são removidos. Enquanto na segunda existe uma melhora em relação à primeira técnica devido à eliminação de detectores redundantes.

Por último é necessário se focar na questão da evolução e no tempo de vida dos detectores. Partindo desse princípio, como o sistema, no qual o IDS irá executar, é finito, ele não pode gerar detectores indefinidamente. Devido a essa preocupação, de acordo com o trabalho de (AYARA et al., 2002) e (GONZÁLEZ; DASGUPTA, 2003), foi proposto que os detectores possuíssem um certo período de vida antes de serem deletados. Já no que se refere à evolução, de acordo com (KIM; BENTLEY, 1999), uma livraria genética foi adicionada ao seu modelo de AIS para redes. Cuja função é a de armazenar dados de detectores, que descrevem padrões anômalos de tráfego na rede. Essa livraria é dinâmica, ou seja, está sempre se atualizando em relação a novos padrões anômalos, e com base nessas atualizações novos detectores são gerados.

Outra abordagem que podemos destacar é a do trabalho de (PARTHASARATHY, 2003), onde os métodos utilizados para criar um IDS foram os de seleção negativa e seleção clonal. O segundo possui a função de simular o processo de seleção natural, em que os detectores com maior afinidade (capacidade de detecção de anomalias) são selecionados. A ideia geral desse método é a de que os detectores serão estimulados pela presença de uma anomalia (antígeno), sendo assim os que tiverem maior afinidade serão selecionados e começarão a se replicar com mutações aleatórias, onde os que tiverem maior afinidade serão selecionados mais uma vez, gerando detectores com afinidade cada vez maior na detecção de anomalias.

Uma abordagem interessante para técnicas inspiradas no sistema imune trata-se da teoria do perigo, em que um sinal de perigo é utilizado para identificar um antígeno (intrusão) que está causando danos, independentemente desse antígeno pertencer ao corpo ou não. Um algoritmo muito utilizado nessa abordagem é o algoritmo de células dendríticas (ACD) (GREENSMITH; AICKELIN; CAYZER, 2005). Nesse algoritmo existem três fases que são inicialização, atualização e agregação. Na fase de inicialização os seus parâmetros são configurados, inicializados e as células dendríticas (CD) se encontram no estado imaturo. Na fase de atualização, as CD são atualizadas em relação aos antígenos e sinais de entrada. Dentre esses sinais de entrada, os dois primeiros são os sinais de perigo (sinais que podem ser produzidos por uma célula que foi danificada por um antígeno), e sinais PAMP, que incrementam o sinal maduro (sinal correspondente a presença de anomalia) de saída das CD. O sinal de saída semi-maduro é incrementado pelo sinal seguro (correspondente a ausência de anomalia). Por último temos o sinal de saída de migração incrementado por todos os sinais de entrada, e o sinal de entrada de inflamação que potencializa o incremento de todos os sinais de saída. Já na fase de agregação, quando as CDs atingem o limiar de migração, elas migram para o linfonodo. No momento em que o linfonodo atinge um determinado número de CD, será calculado o índice de anomalia dos antígenos ou MCAV (*Mature Context Antigen Value*). Caso o MCAV possua um valor maior do que o limiar determinado, o ACD detecta a presença de um invasor. A figura 5 apresenta as fases do algoritmo de células dendríticas.

Uma característica interessante da teoria do perigo é que diferente de outras abordagens em AIS, a anomalia é detectada através do processamento de sinais (Sinal de perigo ou PAMP),

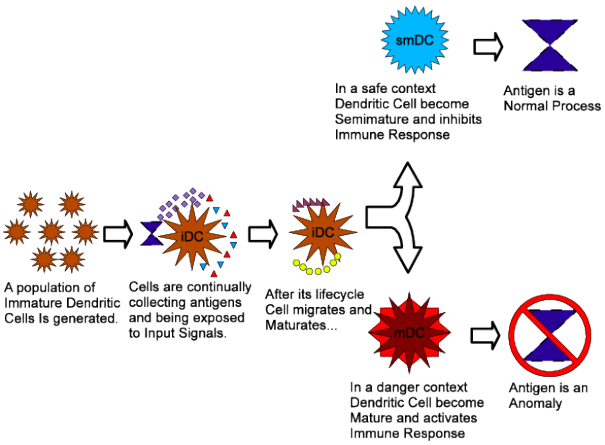


Figura 5 – Algoritmo de Células Dendríticas (SILVA; PALHARES; CAMINHAS, 2012)

e não através da definição de um perfil que representa o padrão de comportamento normal de um sistema. Uma desvantagem é que nessa teoria a intrusão será detectada apenas depois que o sistema já sofreu algum dano.

Outro paradigma dentro da abordagem AIS se trata da teoria de rede imune, em que o algoritmo de rede imune é utilizado. Essa teoria foi proposta por (JERNE, 1974) em 1974, onde nela, o sistema imunológico é visto como uma rede regulada composta de anticorpos e células imune que reconhecem umas às outras mesmo na ausência de antígenos. Dessa forma quando um antígeno é reconhecido ocorre uma reação de proliferação de células imune para que se possa produzir mais anticorpos, já quando um anticorpo ou célula imune é reconhecida ocorre uma supressão na produção de anticorpos (JERNE, 1974). Esse paradigma é baseado no conceito de ativação do sistema imune para patógenos que já foram encontrados. Sendo assim, quando células imunes são repetidamente expostas a um mesmo patógeno, elas podem se defender melhor contra ele em um encontro futuro.

2.5 Correlação de Alertas

Técnicas de correlação de alertas consistem em agregar alertas gerados com a função de construir um cenário de ataque (HUBBALLI; SURYANARAYANAN, 2014). O objetivo principal é fornecer uma visão global ou condensada de ataques de rede (SADODDIN; GHORBANI, 2006). Trabalhos que utilizam essa abordagem geralmente agrupam alarmes originados de diferentes redes ou por diferentes sistemas de detecção na reconstrução do cenário de ataque, uma vez que alertas analisados individualmente geralmente não fornecem informação suficiente para que um IDS possa inferir um resultado (HUBBALLI; SURYANARAYANAN, 2014). Sendo assim, técnicas de correlação auxiliam em detectar a origem de um determinado problema representado por intrusões na rede (SALAH; MACIÁ-FERNÁNDEZ; DÍAZ-VERDEJO, 2013).

De acordo com (HUBBALLI; SURYANARAYANAN, 2014), uma técnica de correlação

de alertas consiste nas seguintes etapas:

- Normalização de alertas: Uma vez que alertas podem se originar de diferentes sistemas de detecção, é necessário que eles possuam um formato em comum. Entre os possíveis formatos, existem os padronizados, como por exemplo *Intrusion Detection Message Exchange Format* (IDMEF) (DEBAR; CURRY; FEINSTEIN, 2007).
- Clusterização de alertas: Essa etapa consiste em agrupar alertas similares em um mesmo grupo, em que atributos como endereço IP de origem, endereço IP de destino, porta de origem, porta de destino, nome do ataque, e serviço, são geralmente considerados. Essa etapa é importante pois a quantidade total de alarmes é reduzida, pois os que possuem alto nível de similaridade entre eles, são fundidos.
- Correlação de alertas: Analisa os grupos formados e os que apresentam alto nível de similaridades são agrupados em um único grupo.
- Reconhecimento de intenção: Identifica o plano que um atacante possui.
- Reportar: Gera uma visão condensada do cenário de ataque para que um administrador possa analisar.

A figura 6 apresenta as etapas seguidas por um algoritmo de correlação de alertas.

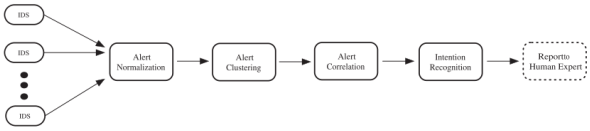


Figura 6 – Técnica de Correlação de Alertas (HUBBALLI; SURYANARAYANAN, 2014)

2.5.1 Correlação de Alertas Baseado em Grafos de Ataque

Um grafo de ataque é uma representação de todos os caminhos através de um sistema que termina em um estado onde um intruso obteve êxito em atingir seu objetivo (JHA; SHEYNER; WING, 2002). De acordo com (SHANDILYA; SIMMONS; SHIVA, 2014), no momento em que um sistema em execução se encontra em um estado indesejado com resultados nocivos, ele representa um cenário de falha. Esse cenário pode ser definido como uma sequência de ações que viola o funcionamento correto do sistema. Sendo assim, o conjunto de todos os cenários de falha podem ser referidos como um grafo de cenário de falha. No caso dessa falha não se encontrar associada a um defeito do sistema, mas a alguma ação de um atacante, esse grafo será denominado grafo de cenário de ataque ou grafo de ataque. Cada caminho representado leva a algum estado indesejado, como por exemplo, um invasor obtendo acesso de administrador a algum servidor de arquivos. A figura 7 apresenta um exemplo de grafo de ataque.

Grafos de ataque podem ser utilizados para detecção, defesa, e análise forense. Na área de detecção eles podem ser utilizados em conjunto com algoritmos de correlação de alertas, permitindo que o IDS possa prever o objetivo dos ataques, a agregação de alarmes para reduzir o volume de informações de alertas que podem ser analisados, e a redução da taxa de alarmes falsos. Na área de defesa esses grafos podem marcar caminhos que um IDS irá detectar, determinar onde posicionar melhor um componente IDS para uma melhor cobertura, explorar melhor as vantagens/desvantagens entre diferentes políticas de segurança ou entre diferentes configurações de *software* ou *hardware*, e identificar os piores cenários de intrusão e dessa forma priorizar estratégias de defesa de acordo. No que se análise forense, o grafo de ataque poderia ser utilizado para encontrar possíveis ações futuras do atacante, após ele ter realizado uma invasão bem-sucedida, além de avaliar os danos causados por ele (JHA; SHEYNER; WING, 2002).

De acordo com (HOMER et al., 2013), uma limitação de grafos de ataque se deve a suposição de que qualquer vulnerabilidade presente nele pode ser explorada. No entanto cada uma delas possui uma probabilidade diferente de ser explorada por um invasor. Para solucionar essa questão foi desenvolvido um sistema de métricas denominado *Common Vulnerability Scoring System* (CVSS) (MELL; SCARFONE; ROMANOSKY, 2006), onde cada vulnerabilidade possui um valor associado ao risco que cada uma representa. De acordo com essa métrica, cada medida de risco está associada a habilidade necessária para se explorar um ponto fraco do sistema, e a informações conhecidas sobre a disponibilidade de uma vulnerabilidade (HOMER et al., 2013).

Abordagens de IDS baseadas em anomalia, entre elas AIS, possuem uma alta taxa de alarmes falsos. Para resolver esse problema, uma possível abordagem poderia ser a utilização de grafos de ataque em conjunto com algoritmo de correlação de alerta. Essa abordagem de correlação é usada entre sistemas de detecção baseados em assinatura para reduzir alarmes falsos e melhorar a precisão. Segundo (HUBBALLI; SURYANARAYANAN, 2014), a técnica de correlação baseada em grafos de ataque (AGC - *Attack Graphs Based Correlation*) parte do princípio de que uma vulnerabilidade em um hospedeiro quando estudado isoladamente pode revelar pouca informação. Por exemplo, um *host* com uma vulnerabilidade de baixo impacto poderia classificar o alerta correspondente como de baixa prioridade. No entanto, um invasor pode usar esse mesmo *host* para alcançar outros sistemas críticos com vulnerabilidades de alto impacto. Isso se deve ao fato de que um atacante pode explorar diversos caminhos existentes para comprometer um sistema. Além disso, um grafo de ataque consiste de informações de vulnerabilidade que apresentam uma relação de interdependência entre os seus vértices de acordo com a topologia da rede.

Na abordagem proposta o grafo de ataque representa todos os caminhos de possíveis ataques a fim de auxiliar em duas etapas do modelo de segurança proposto. O primeira se refere a correlação de alertas, visando um aumento na precisão, e na redução da quantidade de alarmes falsos. A segunda etapa busca selecionar contra-medidas apropriadas a serem executadas em cada cenário de ataque. Para a seleção de contra-medidas são coletadas informações sobre as

vulnerabilidades de cada VM por meio do sistema CVSS.

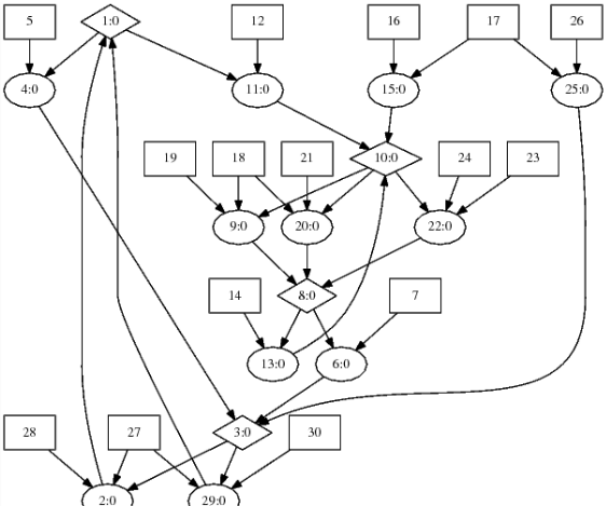


Figura 7 – Exemplo de Grafo de Ataque (OU; GOVINDAVAJHALA; APPEL, 2005)

2.6 Redes Bayesianas

Redes Bayesianas (BN - *Bayesian Network*) consistem em grafos acíclicos dirigidos (DAG - *Directed Acyclic Graph*) com vértices representando variáveis e arestas representando independência condicional entre as variáveis (FRIGAULT; WANG, 2008). Onde um DAG consiste em um grafo que não contém ciclos, ou seja, para qualquer vértice, não há nenhuma ligação dirigida começando e acabando nele mesmo (JENSEN, 1996). Já de acordo com (PEARL, 2014), redes Bayesianas são modelos gráficos que representam explicitamente ligações probabilísticas entre variáveis. Por oferecerem uma forma compacta de codificar todas as relações condicionais das variáveis ou atributos de um sistema, essa abordagem já foi utilizada na área de segurança.

De acordo com (ISLAM et al., 2008), é apresentada uma abordagem onde para cada vértice de um grafo de ataque é atribuído um valor relacionado a probabilidade que uma vulnerabilidade possui de ser comprometida por um ou mais invasores. Dando continuidade ao trabalho anterior, o artigo de (FRIGAULT; WANG, 2008) apresenta uma técnica onde cada probabilidade individual pode ser derivada para se obter a probabilidade condicional, compondo dessa forma uma rede Bayesiana. A probabilidade condicional se refere a probabilidade que cada vértice possui de ser comprometido, considerando todos os outros. Já o artigo de (CHUNG et al., 2013) apresenta um método de prevenção a ataques na nuvem em que a seleção de contra-medidas para mitigar um ataque é realizada por meio da tecnologia de redes SDN. A técnica de BN é utilizada para auxiliar na seleção da melhor contra-medida para cada cenário de ataque por meio das probabilidades condicionais presentes em um grafo de ataque.

De forma semelhante ao trabalho de (CHUNG et al., 2013), a abordagem proposta também utiliza redes Bayesianas para auxiliar na seleção de contra-medidas. No entanto a

diferença se deve ao fato de que o modo de mapeamento dos alertas gerados ocorre por meio dos atributos de origem, destino, e tempo para cada alerta. Onde esse processo de mapeamento, resultando posteriormente na seleção de uma contra-medida para mitigar ataques, será explicado em mais detalhes no capítulo 4.

A seguir, a figura 8 apresenta um exemplo de rede Bayesiana em que os valores que estão na parte de fora de cada vértice correspondem a probabilidade individual de cada um ser comprometido. Os valores no interior estão associados as probabilidades condicionais que cada um deles possuem em relação a probabilidade de sucesso ou falha no momento que um ataque explora suas vulnerabilidades.

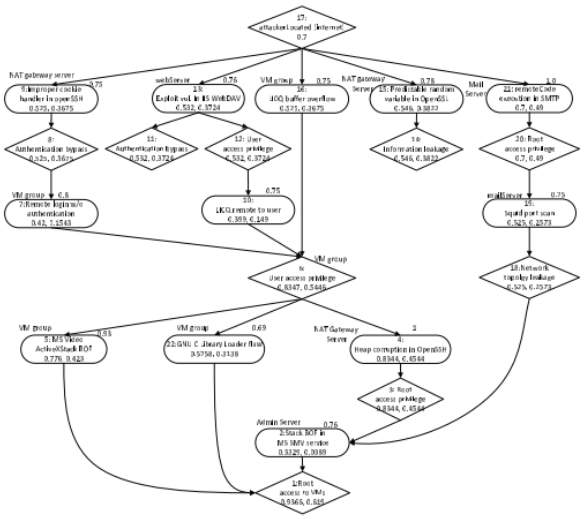


Figura 8 – Exemplo de Rede Bayesiana (CHUNG et al., 2013)

2.7 Redes Definidas por Software (SDN)

Esse tipo de rede possui a característica especial de desacoplar a lógica de controle de implementações fechadas de proprietário, permitindo dessa forma que pesquisadores e proprietários possam desenvolver novas funções ou protocolos de forma mais fácil e flexível (SHIN; GU, 2013). Na arquitetura utilizada por essa tecnologia, o encaminhamento dos segmentos no *data plane* é gerenciado por um *control plane* remoto desacoplado da forma embarcada dos concentradores de rede. A principal função do *data plane* é o encaminhamento de pacotes. No entanto por possuir uma natureza programável, ele ainda possui funções de suporte a aplicações de rede, como por exemplo *deep packet inspection*, onde o conteúdo dos pacotes e de processamento de requisições da rede são analisado(MASOUDI; GHAFFARI, 2016), possibilitando dessa forma tarefas como detecção de anomalia, e engenharia de tráfego. Já o *control plane* é responsável por apresentar uma visão abstrata e centralizada de toda a rede. Ele é composto por controladores baseados em *software* que concentram toda a lógica de programação. Dessa forma, os dispositivos da rede (*data plane*) se tornam ferramentas de encaminhamento de pacotes, que

podem ser programadas por meio de interfaces abertas, como por exemplo, ForCES, e OpenFlow (NUNES et al., 2014).

De acordo com (KREUTZ et al., 2015), SDN pode ser definido como uma arquitetura de rede que possui quatro pilares principais:

- O *control plane*, responsável pela configuração dos nós e da programação dos caminhos utilizados pelo fluxo de dados, é separado do data plane, responsável pelo encaminhamento de pacotes, composto pela rede física, incluindo *Ethernet Switches* e roteadores.
- Decisões de encaminhamento são baseados em fluxo (*flow-based*) ao invés de baseado em destino (*destination-based*) como nos equipamentos convencionais (JAMJOOM; WILLIAMS; SHARMA, 2014).
- O controle lógico (*software*) é movido para uma unidade externa denominada *controller* ou *Network Operation System* (NOS), que analogamente seria o sistema operacional da rede. O NOS é um sistema cliente servidor, que fornece os recursos necessários para facilitar a programabilidade dos dispositivos que segmentam a rede, em um ambiente centralizado, que por sua vez, abstrai a visualização do ambiente de rede.
- As duas principais características das redes SDN são a centralização do controle e a programabilidade. A rede é programada na camada superior do NOS que interagem com os dispositivos (programáveis) na camada subjacente do sistema, no *data plane*.

A arquitetura de uma rede SDN é composta por um controlador (*control plane*) que atua como uma camada intermediária entre a interface de aplicações, denominada *northbound interface* (NBI), e outra interface de infraestrutura (*data plane*), denominada *southbound interface* (SBI). A figura 9 apresenta a arquitetura de uma rede SDN.

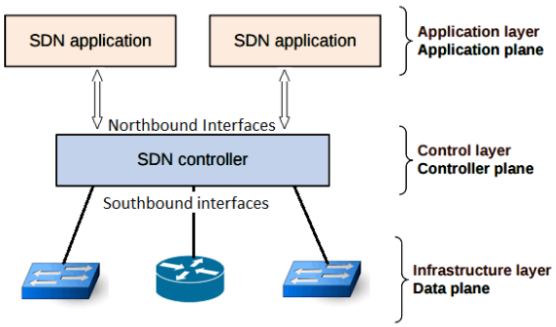


Figura 9 – Arquitetura de Redes SDN (BUILDING..., 2016)

O NBI fornece uma interface de comunicação entre aplicativos da rede e o controlador (SOOD et al., 2015), que inclui uma interface de programação de aplicativos (API) para controle de rede e gerenciamento de programas. Através do NBI, as aplicações podem interagir com

o *hardware* (*switches*) coexistindo e interagindo com outras aplicações e utilizando serviços do sistema, como por exemplo, topologia, descoberta da rede, entre outros. Além disso, essa interface possibilita a integração com plataformas de computação na nuvem do tipo OpenStack, para aplicações de rede, como *firewalls*, balanceadores de carga, sistemas de detecção de intrusão, *backups*. No entanto atualmente não existe uma interface NBI padronizada, embora existam projetos que estão trabalhando nessa questão como o Open DayLight Project (MASOUDI; GHAFFARI, 2016).

A interface SBI é composta por APIs que possibilitam a comunicação entre o *control plane* e o *data plane*. Nela é utilizado o protocolo OpenFlow que se trata da interface mais conhecida entre os *switches* da rede e os controladores. O protocolo OpenFlow foi desenvolvido com o objetivo de padronizar a comunicação entre o *control plane* e o *hardware*. Sendo assim, ele possui uma especificação que viabiliza a migração da lógica de controle dos *swicthes* da rede para os controladores. Entre as funções disponibilizadas por esse protocolo estão análise de tráfego baseada em *software*, controle centralizado, atualização dinâmica de encaminhamento de regras, e abstração de fluxo (MASOUDI; GHAFFARI, 2016). Consequentemente, aplicações baseadas em OpenFlow foram desenvolvidas visando facilitar a configuração ou gerenciamento da rede, adicionar atributos de segurança, virtualizar a rede e centrais de dados, e implantar sistemas móveis (MASOUDI; GHAFFARI, 2016).

Sendo assim, devido as características que essa tecnologia apresenta, ela vem sendo cada vez mais empregada na área de segurança. Um exemplo seria o trabalho de (SEEBER; RODOSEK, 2015) em que é utilizado um conjunto de sistemas de detecção, onde o tráfego entre eles é redirecionado pelo protocolo OpenFlow, ou o trabalho de (HUANG et al., 2015) em que a tecnologia SDN é utilizada para combater ataques do tipo *botnet* que podem provocar ataques de negação de serviço ou de negação de serviço distribuído.

3 Metodologia e Técnica de Pesquisa

A natureza do trabalho científico que está sendo proposto é de caráter bibliográfico, comparativo e experimental. Bibliográfico, porque foi realizada uma pesquisa com o intuito de se obter o estado da arte de IDS baseados em rede dentro da abordagem imunológica para o ambiente da nuvem. Também, em paralelo, foi realizada outra pesquisa visando obter o estado da arte de arquiteturas de segurança para a infraestrutura da nuvem que se utilizam tanto de IDS ou IPS, assim como da tecnologia de redes SDN para contribuir para um ambiente mais seguro. A finalidade se trata de combater diversos tipos de ataques nesse ambiente, como os que se encontram nas quatro categorias estudadas (U2R, R2L, *Probe*, e DoS), que englobam ameaças que exploram vulnerabilidades. Comparativo, porque diferentes versões da abordagem proposta foram comparadas entre si. Uma delas utiliza uma técnica de correlação de alertas baseada em grafos de ataque para uma melhor eficácia na detecção, enquanto outra versão se trata do mesmo IDS sem a utilização dessa técnica. Experimental, porque buscou através de experimentos controlados validar a abordagem proposta, por meio de métricas definidas referentes a eficácia da detecção, como precisão, falso positivo, falso negativo, e taxa de detecção. Caracteriza-se dessa forma, como um estudo quantitativo com alguns aspectos qualitativos no que se refere a análise da seleção de contra-medidas para mitigar os ataques em questão.

Devido à natureza do trabalho proposto ele foi dividido em um conjunto de etapas:

- Realizar uma pesquisa bibliográfica para um melhor entendimento dos assuntos em questão, destacados no referencial teórico;
- Revisão bibliográfica com o objetivo de obter o estado da arte de IDS dentro da abordagem imunológica para o ambiente da nuvem;
- Revisão bibliográfica de *frameworks* de segurança dentro do ambiente da nuvem que utilizam a tecnologia de redes SDN como forma de prover um ambiente mais seguro;
- Desenvolvimento da abordagem de segurança proposta, de forma que ela possua o estado da arte tanto de IDS dentro da abordagem imunológica quanto de *frameworks* de segurança para o ambiente da nuvem que utilizem a tecnologia de redes SDN;
- Realizar uma comparação entre a abordagem de segurança desenvolvida a uma versão similar dela, na qual a diferença entre as duas se deve a utilização de correlação de alertas.

Esse capítulo detalha o processo de pesquisa realizado em busca de trabalhos revelantes nas áreas de sistemas de detecção dentro da abordagem imunológica, e arquiteturas SDN para computação em nuvem. Também são apresentadas as perguntas de pesquisa definidas, as bases

e as palavras chaves utilizadas para cada base, assim como os critérios de inclusão e exclusão utilizados para cada área pesquisada.

3.1 Questões de Pesquisa

Para sistemas de detecção dentro da abordagem imunológica buscou-se identificar em que atributos esses sistemas pretendem melhorar, assim como quais as principais técnicas utilizadas por eles, e quais os tipos de dados analisados. Consequentemente, as seguintes perguntas foram formuladas:

- Quais são as métricas utilizadas para expor os resultados dos NIDS analisados?
- Quais as técnicas ou algoritmos utilizados dentro da abordagem imunológica para NIDS que podem ser utilizados no ambiente da nuvem?
- Que tipo de dados os NIDS analisam?

Enquanto para arquiteturas que utilizam a tecnologia SDN, o objetivo se tratou de identificar de que forma essa tecnologia é utilizada para prover um ambiente mais seguro na nuvem. Sendo assim, a seguinte questão foi definida:

- De que forma a tecnologia de redes SDN é utilizada para prover um ambiente mais seguro na nuvem?

3.2 Palavras Chave da pesquisa

Dentro da área de NIDS, a seguinte *string* de busca foi formulada para a base SCOPUS:

- “Network Intrusion Detection” OR “NIDS” OR “IDS” OR “Intrusion Detection” AND “AIS” OR “Artificial Immune System” AND “Cloud” OR “Distributed” OR “Cloud Computing” OR “Cloud Platform”

Enquanto para Arquiteturas SDN, para as bases SCOPUS, IEEE e Science Direct, foi definida a seguinte *string*:

- “Network Intrusion Detection” OR “NIDS” OR “Intrusion Detection” OR “IDS” OR “Intrusion Prevention” OR “IPS” AND “Cloud Computing” OR “Cloud Platform” OR “Cloud” AND “SDN” OR “Software Defined Network”

3.3 Critérios de Inclusão e Exclusão

Os critérios de inclusão e exclusão definidos para a área de sistemas de detecção em redes dentro da abordagem imunológica foram o seguintes:

- Critérios de Inclusão
  - Artigos, cujo tema se encontre relacionado a NIDS dentro da abordagem imunológica.
  - Artigos, que descrevam arquiteturas, técnicas ou algoritmos dentro da abordagem imunológica para NIDS.
  - Artigos descritos no idioma inglês.
- Critérios de Exclusão
  - Artigos duplicados ou que não estavam disponíveis para leitura online.
  - Artigos cujo NIDS propostos são focados para ambientes de recursos limitados, ou seja, com pouca capacidade de processamento ou memória, como por exemplo: rede de sensores sem fio, pois esses trabalhos não seriam interessantes para o ambiente da nuvem.

Enquanto para arquiteturas SDN para o ambiente da nuvem, os seguintes critérios foram definidos:

- Critérios de Inclusão
  - Artigos que apresentam arquiteturas de segurança na nuvem que se utilizam dos benefícios de redes SDN para prover um ambiente mais seguro.
  - Artigos descritos no idioma inglês.
- Critérios de Exclusão
  - Artigos duplicados ou que não estavam disponíveis para leitura online.
  - Artigos cuja arquiteturas apresentadas não são focadas para o ambiente da nuvem.
  - Artigos que não utilizam a abordagem de redes SDN para prover um ambiente mais seguro.
  - Artigos focados em combater apenas ataques DDoS.

Em relação ao último critério de exclusão para arquiteturas SDN, foram excluídos artigos focados em combater apenas ataques DDoS, pois buscava-se pesquisas voltadas ao combate das quatro classes de ataques estudadas.

3.4 Resultados

Para a área de NIDS dentro da abordagem imunológica, foi realizada uma pesquisa na base de dados SCOPUS, com o objetivo de identificar o estado da arte para sistemas de detecção de intrusão que não estivessem restritos a ambientes de recursos limitados. Ou seja, ambientes com pouco poder de processamento ou memória, como rede de sensores sem fio, por exemplo. Uma vez que um dos objetivos da pesquisa era identificar sistemas de detecção que fossem mais adequados para o ambiente da nuvem. De acordo com os 78 artigos retornados pela SCOPUS, apenas 21 foram selecionados para compor os estudos primários de acordo com os critérios de inclusão e exclusão definidos.

Enquanto para a área de arquiteturas SDN, a *string* de busca definida retornou ao todo 42, 1907, e 553 artigos pelas bases SCOPUS, IEEE,e Science Direct respectivamente. Após a definição dos critérios de inclusão e exclusão, foram selecionados 7 artigos da base SCOPUS, 8 do IEEE, e 5 do Science Direct para compor os estudos primários.

A tabela 2 apresenta as bases de dados utilizadas, o somatório de trabalhos encontrados em cada base de acordo com as *strings* de busca definidas, e o total de artigos selecionados para cada base, dentro de cada área de pesquisa.

Tabela 2 – Quantidade de artigos localizados por base em cada área

Área de Pesquisa	**Base**	**Trabalhos Retornados**	**Trabalhos Selecionados**
NIDS - AIS	SCOPUS	78	21
Arquiteturas SDN	SCOPUS	42	7
	IEEE Xplore	1907	8
	Science Direct	553	5

4 Trabalhos Relacionados

Essa seção descreve os trabalhos selecionados de acordo com a metodologia de pesquisa utilizada. Primeiro serão descritas pesquisas relacionadas a sistemas de detecção dentro da abordagem imunológica, e em seguida arquiteturas de segurança que se utilizam da tecnologia SDN. Posteriormente, o modelo de segurança proposto será comparado com os trabalhos que estão mais próximos de sua abordagem.

4.1 Sistemas de Detecção de Intrusão

Essa seção descreve trabalhos de sistemas de detecção de intrusão para redes que se encontram dentro da abordagem imunológica e não são focados a ambientes de recursos limitados. São descritos trabalhos encontrados na base SCOPUS, que reúne artigos das revistas mais importantes da área de computação, apresentando documentos de editoras ou organizações profissionais importantes como IEEE, ACM, ou Elsevier. Serão apresentados as principais técnicas ou algoritmos utilizados dentro da abordagem imunológica no decorrer dos anos. Em seguida, os artigos mais atuais ou que utilizam paradigmas imunológicos presentes no modelo de segurança proposto, serão descritos em mais detalhes.

4.1.1 SCOPUS

Durante o ano de 2002 apenas um trabalho foi selecionado (HARMER et al., 2002), onde nele é utilizado o algoritmo de seleção negativa. Em sua abordagem, é desenvolvido um AIS adaptativo e baseado em agentes. Em 2005, de acordo com (QIAO; SU; SUN, 2005) os algoritmos utilizados são o de seleção negativa, e algoritmo imune. O algoritmo imune se trata de uma técnica de otimização, responsável pela geração e manutenção do mecanismo de células imunes. Essas células são responsáveis por implementar os mecanismos de defesa nos sistemas de detecção. Sendo assim, esse artigo utiliza uma abordagem de detecção distribuída, e baseada em AIS.

Em 2007 o artigo (LIU et al., 2007) utiliza o algoritmo de seleção negativa, e uma técnica que se baseia na quantidade de concentração de detectores para medir o nível de intrusão. O segundo trabalho do mesmo ano (GOEL; GANGOLLY, 2007) não menciona nenhum algoritmo específico, no entanto utiliza uma abordagem imunológica que não utiliza o conceito de padrão normal ou anômalo. Além disso, nele é utilizado o conceito de epidemiologia, focado em estudar como as doenças se propagam no decorrer do tempo. O terceiro trabalho de 2007 (LUTHER et al., 2007) utiliza os algoritmos de seleção negativa e seleção clonal, além de se basear na abordagem de seleção positiva, em que detectores que casam com padrões normais não são

descartados.

No ano de 2009 o artigo (YANG et al., 2009) utiliza apenas o algoritmo de seleção clonal em conjunto com uma técnica de cálculo da seriedade de uma intrusão. Essa técnica auxilia na avaliação da segurança da rede. Outro artigo no mesmo ano (JIANG et al., 2009) utiliza os algoritmos de seleção clonal e seleção negativa em conjunto com uma técnica de vacina. Nessa técnica, informações sobre intrusões capturadas de uma rede são encapsuladas e enviadas para uma rede vizinha. Dessa forma a rede vizinha passa a possuir informações sobre novas ameaças, auxiliando no processo de detecção. Em (ALI; AIB; BOUTABA, 2009) é utilizado a abordagem de seleção negativa, no entanto para a geração dos detectores é utilizado o algoritmo de geração aleatória Quasi-Random. Dessa forma é possível obter uma melhor cobertura do espaço de padrões dentro do conjunto *nonself*, ou seja, conjunto no qual se encontram os padrões anômalos ou intrusões. Já em (JIANG; CHANG, 2009) são utilizados os algoritmos de seleção negativa e seleção clonal em conjunto com a técnica de vacina.

Em 2010 o trabalho de (YU; WANG, 2010) não cita nenhum algoritmo específico, no entanto ele utiliza um modelo de AIS em que o sistema de detecção é distribuído em diversos nós dentro de uma rede, no qual eles possuem um sistema de controle central. O foco desse modelo é diferenciar o *self* (padrões normais) do *non-self* (padrões anômalos). Outro artigo do mesmo ano (JIANG; CHANG, 2010) utiliza a abordagem de agentes em que eles cooperam entre si. Nesse trabalho os algoritmos de seleção negativa e seleção clonal são utilizados em conjunto com a técnica de vacina. Em (AKYAZI; UYAR, 2010) são utilizados os algoritmos de seleção negativa e seleção clonal em conjunto com o algoritmo jREMISA (HAAG et al., 2007). No qual o último se trata de um algoritmo evolucionário inspirado em AIS, que procura melhorar as taxas de detecção e de alertas falsos. Ainda no ano de 2010, o trabalho (HOSSEINPOUR et al., 2010) utiliza o algoritmo de seleção negativa dentro de uma arquitetura multicamada distribuída. Essa arquitetura possui o propósito de melhorar o desempenho e eficiência da detecção.

No ano de 2011, de acordo com (WANG; SUN, 2011), um algoritmo denominado *Ant Algorithm* é utilizado dentro da abordagem imunológica com o objetivo de gerar detectores de qualidade em um menor tempo, se comparado com outras técnicas de geração aleatória. Ainda no mesmo ano, o trabalho de (RUIRUI et al., 2011) apresenta um novo método dentro do paradigma de Teoria do Perigo, no entanto não cita nenhum algoritmo específico. Já em 2012, de acordo com (UWAGBOLE; BUCHANAN; FAN, 2012), é utilizado o paradigma de teoria do perigo em que o algoritmo de células dendríticas trabalha em conjunto com a técnica de vacina.

Em 2013 o artigo (ELHAJ; HAMRAWI; SULIMAN, 2013) utiliza uma abordagem de AIS que se encontra dividida em duas camadas: inata e adaptativa. Apenas a parte inata foi testada. Além disso o artigo não cita nenhum algoritmo específico dentro da abordagem imunológica. No entanto, ele utiliza a técnica de *fuzzy logic*, onde ela é utilizada para a tomada rápida de decisões em situações de incerteza, como por exemplo: dados vagos ou incompletos.

Em 2017, o trabalho de (AHMAD; IDRIS; KAMA, 2017) utiliza o algoritmo de células

dendríticas. Nesse artigo é desenvolvido um protótipo de IDS para a nuvem inspirado no mecanismo de células dendríticas.

4.1.2 MAIS-IDS: A Distributed Intrusion Detection System Using Multi-Agent AIS Approach

Esse trabalho (SERESHT; AZMI, 2014) se baseia em um sistema de detecção composto por agentes distribuídos, que utiliza os paradigmas de seleção negativa, seleção clonal, e rede imune. O sistema proposto possui a função de analisar tanto tráfego de rede quanto configurações de sistema, onde o tráfego de rede é analisado a nível de máquina virtual. Dessa forma, o paradigma de seleção negativa é utilizado para remover agentes com baixa eficácia. A seleção clonal realiza cópias do melhor entre eles, e busca melhorar a sua população. Já o paradigma de rede imune é utilizado para colaboração entre agentes, auxiliando na redução de alarmes falsos.

Esse sistema foi testado com a base de dados NSL-KDD (University of New Brunswick, 2018) para 397 registros de tráfego de rede, e obteve uma média de 89% de precisão para 20 execuções independentes. Como trabalhos futuros, o artigo propõe o teste do sistema em ambientes mais realistas.

4.1.3 Distributed Network Intrusion Detection System: An Artificial Immune System Approach

Esse trabalho (IGBE; DARWISH; SAADAWI, 2016) desenvolveu um *framework* de sistema de detecção de intrusão de rede distribuído, baseado na abordagem imunológica. A partir dele é proposto um mecanismo imune adaptativo por meio de aprendizagem de máquina não supervisionada, com o objetivo de classificar o tráfego de rede em normal ou anômalo. Nessa arquitetura, o algoritmo de seleção negativa (NSA) é utilizado em conjunto com algoritmo genético (GA - *Genetic Algorithm*) cuja função é a de auxiliar na geração de detectores.

Essa abordagem (IGBE; DARWISH; SAADAWI, 2016) consiste em agentes autônomos que se comunicam entre si, no qual cada agente corresponde a um NIDS diferente. Dessa forma cada um deles executa independentemente o algoritmo NSA para obter os seus conjuntos individuais de regras. Essas regras ou detectores serão utilizados para classificar o tráfego em normal ou anômalo. Devido a capacidade de comunicação que eles possuem entre si, é possível o compartilhamento de regras de detecção entre os NIDS. Essa característica poderia amenizar situações de ataque de dia zero, em que um determinado NIDS se depara com algum tipo de ataque pela primeira vez, onde devido a sua capacidade de compartilhamento, o ataque em questão poderia ser detectado.

Dentro do *framework* desenvolvido, o algoritmo proposto (NSA-GA) (IGBE; DARWISH; SAADAWI, 2016), que utiliza as técnicas de seleção negativa, e algoritmo genético, é comparado com outras três técnicas (Naïve Bayes, J48, e SVM) (University of Minnesota Duluth, 2018),

para dados de treinamento do NSL-KDD (University of New Brunswick, 2018), e obteve a melhor taxa de detecção. Sendo assim, posteriormente, esse sistema é simulado para ocorrências de dia zero, onde ele é apresentado a ataques novos. Nesse cenário, esse sistema detectou 207 de um total de 300 novos ataques, para dados de teste do NSL-KDD. No entanto o trabalho em questão ainda poderia ser melhorado utilizando outras técnicas dentro da abordagem AIS, com a finalidade de melhorar a comunicação entre os agentes.

4.1.4 A Population-based Incremental Learning Approach with Artificial Immune System for Network Intrusion Detection

Entre os trabalhos mais recentes dentro da abordagem imunológica, podemos citar (CHEN; CHANG; WU, 2016). Nesta pesquisa, a abordagem imunológica é combinada com PBIL (*population-based incremental learning*) e CF (*collaborative filtering*). O primeiro refere-se a uma técnica de aprendizagem incremental com o objetivo de melhorar a população de detectores. Já o segundo corresponde a um método de classificação para auxiliar na detecção de novas intrusões. Dessa forma, o objetivo desse trabalho se trata de melhorar a eficácia da classificação de dados estatísticos do tráfego de rede, e da validação de cartões de crédito.

A abordagem proposta (CHEN; CHANG; WU, 2016) obteve uma média de precisão de 93.24% quando testado com 31.124 registros de dados estatísticos da rede presentes na base de dados KDD99 (University of California, Irvine, 2018). No entanto, em relação a trabalhos futuros, essa pesquisa busca implementar mecanismos para identificar o tipo de anomalia detectada (R2L, U2R, *Probe*, e DOS). Assim como implementar outras técnicas que possam auxiliar na detecção, como o mecanismo de seleção negativa, por exemplo.

4.1.5 An Immune Inspired Unsupervised Intrusion Detection System for Detection of Novel Attacks

Nesse artigo (JHA; ACHARYA, 2016) é proposto um sistema de detecção que utiliza os algoritmos de *T-cell*, e *B-cell*, para analisar tráfego de rede. O primeiro identifica dados de tráfego suspeito, que posteriormente são enviados para o algoritmo de *B-cell* para identificar ataques. Sendo assim, o algoritmo de *B-cell* utiliza dados de cada registro do tráfego, informações fornecidas pelo algoritmo *T-cell*, e seu próprio método de reconhecimento de atributos de cada registro para auxiliar na detecção de tráfego anômalo.

Esse sistema foi testado com a base de dados KDD99 (University of California, Irvine, 2018), onde foram realizados dois experimentos. No primeiro, o IDS foi testado para as categorias de ataques de DoS, e *Probe*, obtendo uma precisão de 77.8%. Já o segundo experimento simulou um cenário de ataque de dia zero, onde a primeira base de dados possuía diversos ataques nas categorias citadas, enquanto os dados de testes possuíam apenas o ataque de varredura de ip, que

se tratava de uma ameaça ainda não analisada pelo sistema. Para esse experimento foi obtida uma precisão de 98%.

4.2 Arquiteturas de Segurança

Essa seção descreve arquiteturas de segurança para a nuvem que se utilizam de benefícios fornecidos por redes SDN visando prover um ambiente mais seguro. Serão apresentados trabalhos encontrados nas bases SCOPUS, IEEE e Science Direct. Em seguida as arquiteturas que estiverem mais próximas da abordagem de segurança proposta serão descritas em mais detalhes.

4.2.1 SCOPUS

O trabalho de (SEEBER; RODOSEK, 2015) utiliza uma abordagem composta por um OpenFlow IDS, responsável por analisar contadores OpenFlow, e um controlador SDN. Sendo assim, o IDS detecta uma ameaça, enquanto o controlador SDN processa o evento gerado pelo OpenFlow IDS em conjunto com o fluxo de dados associado ao evento. Dessa forma, o controlador pode decidir qual contra-medida será selecionada. Em seguida, as regras de OpenFlow são atualizadas, e aplicadas ao OpenFlow *Switch* associado ao controlador SDN. Todo esse processo é conhecido como um ciclo de detecção. Nessa arquitetura podem existir múltiplos ciclos, onde dentro dela o controlador SDN é descentralizado, e consequentemente ele se encontra distribuído em diversos pontos. Em cada um desses pontos, o controlador estará associado a um OpenFlow *Switch* e OpenFlow IDS diferentes. Na primeira etapa dessa arquitetura cada unidade do controlador distribuído decide de forma autônoma que ações executar. Em seguida uma conexão entre eles é estabelecida para melhorar a capacidade de detecção, e possibilitar a implementação de um controlador distribuído. A figura 10 apresenta a arquitetura descrita.

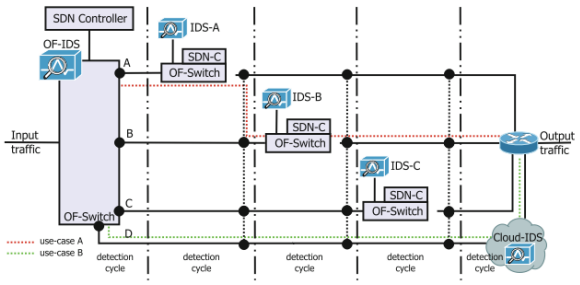


Figura 10 – Arquitetura SDN - (SEEBER; RODOSEK, 2015)

No trabalho de (XING et al., 2014), o tráfego gerado pelas máquinas virtuais são analisados pelo Snort e dessa forma os alertas gerados por ele são passados para um arquivo de *log*. Em seguida os alertas são analisados para extrair apenas informações importantes (tipo de ataque, IP de origem, IP de destino, porta TCP). Posteriormente, através dessas informações serão geradas novas regras de OpenFlow para que se possa reconfigurar a rede através da

tecnologia SDN (topologia, cabeçalho de pacote, parâmetros de qualidade de *software* (QoS), redirecionamento/isolamento de tráfego, filtragem de tráfego, bloqueio de portas).

A arquitetura de (HUANG et al., 2015) apresenta três mecanismos de segurança denominados: mecanismo de bloqueio de *botnet/malware*, mecanismo de filtro de escaneamento, e mecanismo *honeypot*. No primeiro, o Snort irá detectar ataques *botnet/malware*, no segundo, ameaças do tipo DoS ou DDoS serão detectadas através do auxilio de *honeypots*. Onde *honeypots* se tratam de armadilhas com o intuito de provocar negação de serviço (DoS/DDoS), possibilitando a detecção desses ataques pelo Snort. Enquanto o filtro de *scanner* possui um algoritmo para detectar escaneamentos ilegais. Através desses três mecanismos, em qualquer tentativa de ataque, será gerado um alerta que será passado para o controlador de rede SDN. Dessa forma, a comunicação entre os programas de detecção (Snort e algoritmo de escaneamento) e o controlador, ocorre através de um algoritmo proposto, denominado *Cooperative Threat Defending Algorithm* (CTDA). Sendo assim, após a passagem do alerta para o controlador, será instalada uma regra OpenFlow para que se inicie uma contra-medida para o ataque identificado, e em seguida o tráfego malicioso será bloqueado ou isolado, e qualquer tráfego subsequente que apareça da mesma fonte também será bloqueado. Na arquitetura proposta, o Snort se encontra em um servidor fora da nuvem privada, enquanto os programas de *honeypot* e filtro de escaneamento se encontram em cada máquina virtual.

O trabalho de (LI; LIU; LIN, 2016) propõe uma arquitetura onde a ideia é monitorar o trafego entre as máquinas virtuais que pertencem a diferentes domínios de segurança, uma vez que IDS como o Snort não são capazes de desempenhar essa tarefa. Dessa forma, o tráfego entre as máquinas virtuais é exportado e encaminhado para IDS físicos através da abordagem SDN pelo protocolo OpenFlow. Sendo assim, cada IDS físico é responsável por monitorar o tráfego de saída e entrada de cada domínio de segurança que podem se encontrar no mesmo servidor ou espalhados em mais de um servidor, uma vez que cada domínio é formado por um conjunto de máquinas virtuais. Consequentemente, os IDS utilizados apenas analisam o tráfego entre domínios de segurança diferentes, os que pertencem ao mesmo domínio são excluídos.

De acordo com esse artigo (HA et al., 2016a), é proposto um método de ajuste na análise de pacotes. Onde devido a tecnologia SDN, é possível analisa-los por meio de *switches* OpenFlow através da duplicação dos pacotes e do seu redirecionamento para o IDS para inspeção. No caso de pacote malicioso, um alarme é enviado para o controlador SDN. Onde baseado no resultado da analise do IDS e no status atual dos *switches*, o controlador SDN pode reconfigurar a rede para se defender de ataques. Devido a limitações da capacidade de análise de pacotes do IDS, um algoritmo para determinar a taxa de análise dos *switches* da rede é desenvolvido para garantir que o total de pacotes analisados continue abaixo da capacidade máxima do IDS. Ao mesmo tempo, o algoritmo minimiza a taxa de pacotes maliciosos perdidos.

Esse trabalho desenvolveu um *framework* (XIA; CHEN; XU, 2016) que procura adaptar IDS para o ambiente da nuvem, para detectar ataques de dentro desse ambiente. Sua ideia central

é dinamicamente balancear a coleta de dados e custo computacional de acordo com a utilização de recursos na nuvem. Além disso, possui o objetivo de aumentar a probabilidade de que um IDS capture tráfego de ataque, impedindo que tanto o IDS quanto o *datacenter* se sobrecarreguem. No mecanismo de processamento local, a tecnologia SDN é utilizada em cada *swicth* virtual (Open vSwitch), presente em cada servidor físico para filtragem (Para diminuir a sobrecarga de comunicação devido a pacotes desnecessários), e coleta de dados estatísticos do tráfego da rede, para depois encaminhar o tráfego para o IDS. Já no mecanismo de processamento global são computados estatísticas globais do tráfego de rede, através do *software* Apache *Storm* (Responsável pela contagem de fluxos e pelo cálculo do atributo de entropia). Dessa forma, dados estatísticos de pacotes e fluxos são coletados tanto a nível global, quanto a nível local. Em seguida esses dados são enviados para um controlador central, para determinar tanto a taxa de análise dos pacotes, como a prioridade deles. A partir dele são definidas políticas para decidir quais pacotes serão analisados. No controlador central existe um algoritmo para decidir quais pacotes serão analisados. Ele possui duas funcionalidades: Ajuste dinâmico da taxa de análise de acordo com a utilização de recursos, e seleção dos fluxos mais suspeitos para que um IDS possa analisar. Sendo assim, o controlador decide quais pacotes serão processados pelos demais componentes da arquitetura.

De acordo com (QIU; ZHANG; REN, 2017), é proposto um mecanismo denominado *Global Flow Table* (GFT). Baseado no mecanismo de *flow table* de redes SDN, fundamental para lidar com pacotes de dados em SDN *switches*. O GFT fornece uma visão geral de toda a rede. A partir dele são gerados e armazenados caminhos completos de cada fluxo, contendo informações relacionadas a direção, *switches* que eles percorrem, portas que os encaminharam, e volume de tráfego de toda a rede. O GFT auxilia no mecanismo de detecção a partir da identificação de fluxos suspeitos ou nós da rede, onde os dois serão reportados para dispositivos mais sofisticados de detecção, para que se possa confirmar o ataque.

4.2.2 IEEE

De acordo com (SAYEED; SAYEED; SAXENA, 2015), foi desenvolvido um *firewall*, cuja função se trata de filtrar pacotes de um controlador de rede SDN baseado em java, denominado Floodlight (FLOODLIGHT, 2018). A partir do *firewall*, são extraídos padrões e geradas regras de associação dos pacotes filtrados por meio do algoritmo Apriori (HAN; PEI; KAMBER, 2011). Essas regras são comparadas com outras predefinidas, onde na ocorrência de algum desvio de acordo com as regras predefinidas, o IDS irá gerar um alerta, e consequentemente será atualizado.

Já de acordo com (LE et al., 2015), foi desenvolvida uma arquitetura de detecção e prevenção que recebe dados dos *switches* Openflow. Nela, primeiramente, os dados passam por um módulo de extração de atributos para que o melhor conjunto de atributos possa ser selecionado. Dessa forma, esse processo irá facilitar o módulo de classificação de tráfego, onde o

algoritmo de aprendizado de máquina denominado C4.5 (QUINLAN, 2014) é utilizado para esse fim. No caso de alguma ameaça detectada, o módulo de detecção irá interagir com o controlador SDN para que novas regras sejam implantadas nos *switches* Openflow. Sendo assim, a arquitetura proposta pode se encontrar no modo de treinamento, no qual o algoritmo C4.5 treina com dados rotulados enviados pelo módulo de treinamento, que por sua vez já recebeu esses dados com os atributos selecionados pelo módulo de extração. No modo normal, o algoritmo C4.5 recebe dados não-rotulados diretamente do módulo de extração. Por fim, existe o módulo de *log*, onde informações de ataques são salvas, e dados de treinamento são gerados. A figura 11 apresenta a arquitetura descrita.

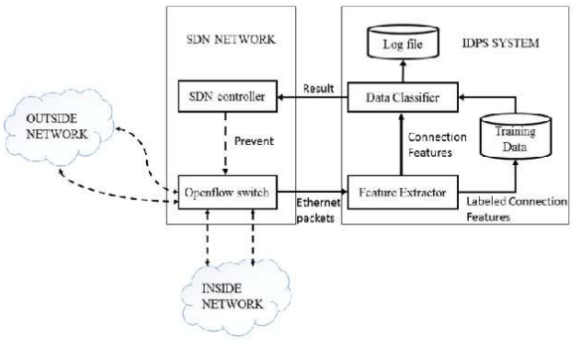


Figura 11 – Arquitetura SDN de Detecção e Prevenção - (LE et al., 2015)

O trabalho de (HA et al., 2016b) se foca em distribuir o tráfego entre os IDS visando melhorar o desempenho da detecção e balancear a carga de dados analisados entre cada sistema de detecção. A ideia central se trata de agrupar fluxos de tráfego malicioso que pertencem a um mesmo ataque para o mesmo IDS analisar. Pois muitos algoritmos de detecção utilizam regras para verificar se o número de pacotes que percorrem um determinado destino em uma determinada unidade de tempo excede um certo limiar definido. Consequentemente, é mais fácil verificar se o limiar foi excedido caso todos os fluxos de tráfego pertencentes a um mesmo ataque sejam redirecionados para um único IDS analisar. Já em relação ao balanceamento das cargas, no caso em que o tráfego da rede não é gerado uniformemente, ele pode ser redirecionado para qualquer IDS. Resultando em uma carga balanceada para cada sistema de detecção. Para o agrupamento dos fluxos, a proximidade do caminho de roteamento de cada um deles é observada. Dessa forma os fluxos são agrupados de acordo com a técnica de *gravity clustering* (INDULSKA; ORLOWSKA, 2002). Esse trabalho apresentou resultados em que a técnica utilizada obteve melhor desempenho na detecção de ataques e balanceamento de cargas do que técnicas que agrupam fluxos aleatoriamente entre os IDS.

No artigo de (JEONG et al., 2014) é proposta uma arquitetura escalável onde são explorados atributos da tecnologia SDN para melhorar a detecção de ataques e ajustar a taxa de análise de cada IDS. O propósito desse ajuste se trata de limitar a taxa de inspeção de um IDS para que ele não exceda sua capacidade, evitando uma degradação significativa no seu

desempenho. Para atingir esse objetivo é proposto um método de ajuste onde a taxa de análise em cada *switch* da rede é ajustado visando minimizar a probabilidade de que pacotes maliciosos não sejam detectados quando encaminhados para os IDS. Nessa arquitetura cada *switch* na rede pode analisar tráfego malicioso e redirecionar o tráfego para a sua inspeção pelos sistemas de detecção presentes. Além disso, é utilizado um controlador Floodlight (FLOODLIGHT, 2018) que utiliza o protocolo OpenFlow para um controle flexível dos fluxos de dados, permitindo dessa forma uma inspeção de tráfego escalável. Esse trabalho foi testado com outro algoritmo simples onde o fluxo de dados é analisado uniformemente em cada *switch* apenas se o conjunto desses fluxos não excede a capacidade de inspeção do IDS. No entanto, o algoritmo proposto obteve um resultado melhor em relação a detecção, pois ele aumenta gradativamente a taxa de análise dos *switches* em que pacotes maliciosos foram previamente identificados.

No trabalho de (KARMAKAR; VARADHARAJAN; TUPAKULA, 2017) é proposta uma arquitetura baseada em políticas de segurança para mitigar ataques. As políticas especificadas se tratam de regras que determinam os fluxos ou caminhos que pacotes podem percorrer em uma rede e sob que condições eles podem ser percorridos. A arquitetura proposta pode fazer parte da estrutura do controlador SDN ou pode se tratar de uma aplicação executando sobre ele. Dentro dela existe o repositório de política, onde as regras de fluxo se encontram, e repositório de topologia, que fornece informações sobre a topologia da rede. Existe também o componente denominado *Evaluation Engine*, responsável por avaliar o fluxo de tráfego contra regras relevantes para cada fluxo. *Policy Manager*, no qual o objetivo se trata de determinar as regras de fluxo que serão encaminhadas para o *Enforce Module*. Já no *Enforce Module* as regras definidas pelo *Policy Manager* são aplicadas. Por último, existe o *Packet Handle*, onde a autenticidade dos pacotes é averiguada, e as políticas definidas são aplicadas em cada *switch*. Esse trabalho foi testado para ataques contra o controlador de rede SND, e para dispositivos SDN dentro do domínio da rede. Para medir seu desempenho foram analisadas sua taxa de transferência, utilização de CPU e de memória *Heap*. Onde a taxa de transferência reduzia a medida em que novas políticas eram implantadas, e obtinha o seu maior valor na ausência delas. Isso deve ao fato de que para as políticas implantadas, apenas os fluxos permitidos são instalados, então como consequência, a utilização de CPU e memória *Heap* são maiores na ausência de regras. A figura 12 apresenta a arquitetura descrita.

No artigo (YE et al., 2016) é proposto um modelo de detecção anômalo de comportamento baseado em computação na nuvem. Nele a tecnologia OpenFlow é utilizada para redirecionamento do tráfego. Dessa forma, o tráfego das máquinas virtuais presentes em um mesmo servidor físico pode ser redirecionado para a arquitetura proposta. Possibilitando que o tráfego inter-VM (Máquinas virtuais que pertencem a domínios diferentes) possa ser monitorado e gerenciado. A figura 13 demonstra como o tráfego é redirecionado para a arquitetura proposta.

Nessa arquitetura (YE et al., 2016) existe o VM *profile module*, responsável por armazenar e gerenciar perfis de VM baseado na análise de tráfego. A partir dessa análise são

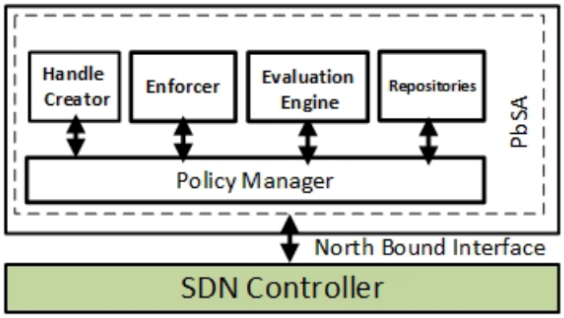


Figura 12 – Arquitetura baseada em Políticas de Segurança - (KARMAKAR; VARADHARAJAN; TUPAKULA, 2017)

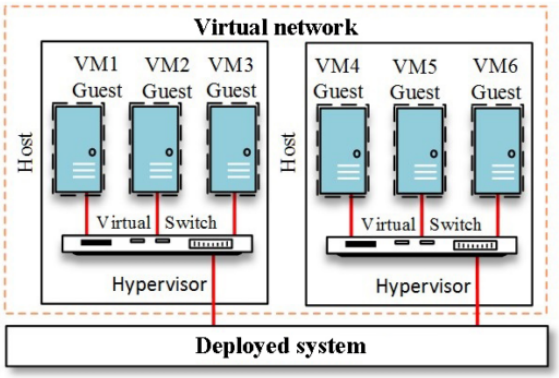


Figura 13 – Redirecionamento do Tráfego - (YE et al., 2016)

obtidos estados de comportamento de aplicações que são utilizados na construção dos perfis das VM. Existe também o módulo Snort em que é utilizada uma técnica de detecção baseada em assinatura para detectar ataques conhecidos. A partir dele, não apenas o tráfego é analisado como também o volume de dados que precisa ser processado no próximo módulo é reduzido. Em seguida existe o componente *Data process*, em que são recebidas informações de pacotes ou fluxos de tráfego. Dentro dele estão incluídos análise dos dados de pacotes, reorganização de sessão de fluxo, estatísticas de pacotes, estatísticas de fluxo, e uma interface de acesso de dados. Esse módulo também possui a função de preparar o conjunto de dados que recebe para que eles possam ser utilizados por outros módulos. Já o componente *Application Behaviour Analysis* se divide em duas etapas. A primeira visa identificar aplicações das máquinas virtuais. A segunda fase busca identificar comportamento anômalo de cada aplicação, e para isso é utilizada uma técnica de séries de tempo (BOX et al., 2015). Esse módulo também utiliza o algoritmo de classificação AdaBoost (FREUND; SCHAPIRE et al., 1996). Uma vez que são obtidos os resultados de detecção do Snort e do componente *Application Behaviour Analysis*, eles são salvos como registros de anomalia no *Anomaly record*. Em seguida, para melhorar a precisão da detecção é utilizada uma técnica de análise em profundidade sobre os registros obtidos. Para isso, um algoritmo de analise de decisão é proposto pelo artigo, em conjunto com um classificador Naive Bayesian (MOBASHER, 2005). A figura 14 apresenta o modelo de detecção descrito.

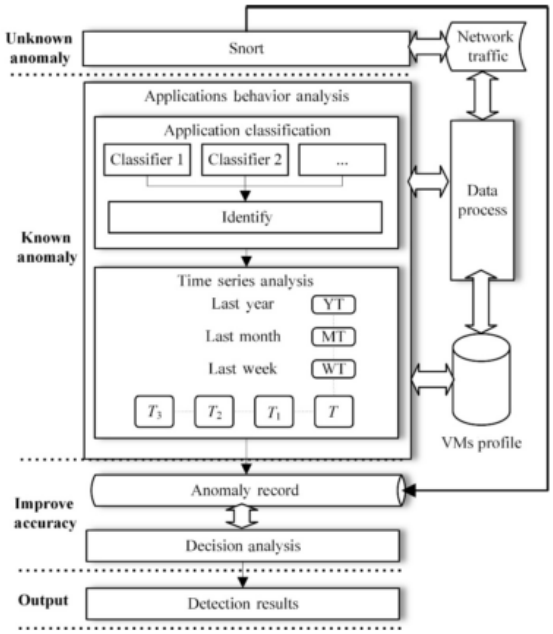


Figura 14 – Modelo de Detecção de Comportamento Anômalo - (YE et al., 2016)

De acordo com o artigo (SINIARSKI et al., 2016), é apresentada uma arquitetura SDN na qual é integrada uma plataforma de análise de *log* em tempo real para a nuvem, denominada LogEntries (LOGENTRIES, 2017), com a função de detectar ataques ou sobrecarga da rede. Essa plataforma é responsável por coletar *log* ou registros de máquinas pertencentes a uma rede, *switches* OpenFlow, e controladores SDN. Uma vantagem referente a ela se deve ao fato da análise dos registros ocorrer em tempo real, uma vez que eles são analisados e processados assim que são transmitidos para o servidor. Isso permite a imediata visualização de notificações, alertas ou eventos. Sendo assim, na arquitetura proposta um agente de *log* é responsável por coletar *logs* de controladores SDN, e da performance do servidor que os executa através de seu monitoramento. Esse agente também é responsável por coletar registros de máquinas que executam componentes da rede na camada de infraestrutura da nuvem. Os *logs* coletados são encaminhados para a plataforma de análise de *logs*, e em seguida processados. Isso permite que administradores de rede possam visualizar, correlacionar, ou analisar *logs* da rede SDN em tempo real. Dessa forma, a partir da capacidade de notificações da plataforma de análise, o componente *User Feedback Channel* apresenta as informações de forma que possibilita notificar o administrador de todos os eventos críticos que podem afetar o desempenho da rede, dos possíveis ataques que possam ocorrer, e da saúde da rede como um todo. A arquitetura descrita é apresentada na figura 15.

4.2.3 Science Direct

O trabalho de (GIOTIS et al., 2014) apresenta uma arquitetura que utiliza o protocolo OpenFlow em conjunto com o método de monitoramento de tráfego denominado sFlow (PHAAL;

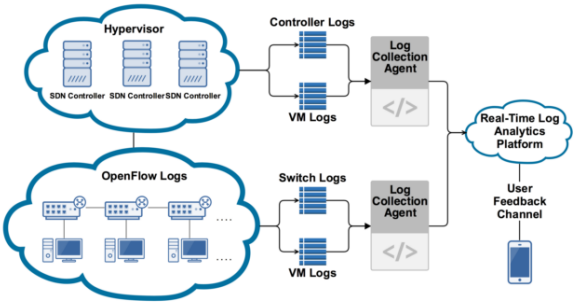


Figura 15 – Arquitetura de Monitoramento em Tempo Real - (SINIARSKI et al., 2016)

PANCHEN; MCKEE, 2001). A vantagem desse método se deve ao fato de evitar a utilização de contadores OpenFlow para coleta de dados estatísticos do tráfego. Permitindo que a contagem dos dados se realize por meio de controladores OpenFlow ao invés de *switches*. Isso resulta na redução da utilização de recursos como CPU, e cache de fluxo de entrada. Além disso, como o método sFlow não requer análise individual de cada fluxo, entradas de fluxo correspondentes a fluxos agregados podem ser redirecionados sem afetar o desempenho do sistema de detecção. A utilização de fluxos agregados pode ser necessária em casos de grande aumento de tráfego (Ataques DDoS), uma vez que aumenta o número de entradas de fluxo nas tabelas dos *switches*. Essa arquitetura é composta por três componentes: coletor, *anomaly detection*, e *mitigation detection*. O primeiro é responsável por coletar dados para que se torne possível a detecção de anomalia por meio dos fluxos de tráfego. O segundo recebe dados do primeiro módulo para realizar a detecção de anomalia em intervalos regulares de tempo, compondo janelas de tempo. Esse módulo pode ser implementado com diversos algoritmos de detecção, no entanto para esse trabalho é utilizado um algoritmo baseado em entropia (LAKHINA; CROVELLA; DIOT, 2005). Já o último componente é responsável por mitigar os ataques através do bloqueio de tráfego por meio da abordagem SDN.

O artigo (SU et al., 2015) apresenta um sistema denominado CeMon, cuja função se trata de coletar estatísticas de fluxo do tráfego em redes SDN. O objetivo do trabalho é apresentar um sistema de alta precisão na coleta de estatísticas de fluxo no qual a sobrecarga de análise dos dados coletados é a mínima possível. A partir dele, outros aplicativos de medição são capazes de invocar CeMon para coletar dados de fluxo a uma sobrecarga mínima. Sendo assim, esse sistema pode servir de suporte para tarefas de monitoramento, como utilização de *links*, estimativa de matriz de tráfego, e detecção de anomalias. Sua arquitetura consiste no *flow event handler*, responsável por receber mensagens de chegada/expiração dos *swicthes* e encaminhá-las para o *routing module*, e *flow state tracker*. Onde o primeiro é responsável por calcular o caminho de roteamento de acordo com as políticas definidas pelo administrador, enquanto o segundo mantém os fluxos ativos na rede. Dessa forma, os dois últimos componentes reportam conjuntos de fluxos ativos em conjunto com seus caminhos de roteamento para o *polling scheme optimizer*. Onde a função do *polling scheme optimizer* se trata de computar o menor custo de análise dos fluxos e

enviar o resultado para o *stat collector*. Para realizar essa função, são utilizados dois esquemas de análise, o *Maximum Coverage Polling Scheme* (MCPS), e *Adaptive Fine-grained Polling Scheme* (AFPS). O primeiro é responsável por coletar estatísticas de todos os fluxos ativos dos *switches*. Dessa forma todos os esquemas de analise são encontrados, possibilitando a escolha do esquema de menor custo. Já o AFPS se trata de um método complementar que coleta estatísticas de um subconjunto de fluxos ativos. Ele ajusta a frequência de análise, aumentando ela quando o tráfego está mais congestionado, e reduzindo a frequência quando o tráfego se encontra menor. Em seguida, o *stat collector* realiza a análise das estatísticas de fluxo dos *switches*, e se encarrega da resposta recebida. Por último, o *flow stat aggregator* agrupa informações sobre as estatísticas de fluxo e fornece interface para aplicações de monitoramento. A figura 16 apresenta a arquitetura denominada CeMon.

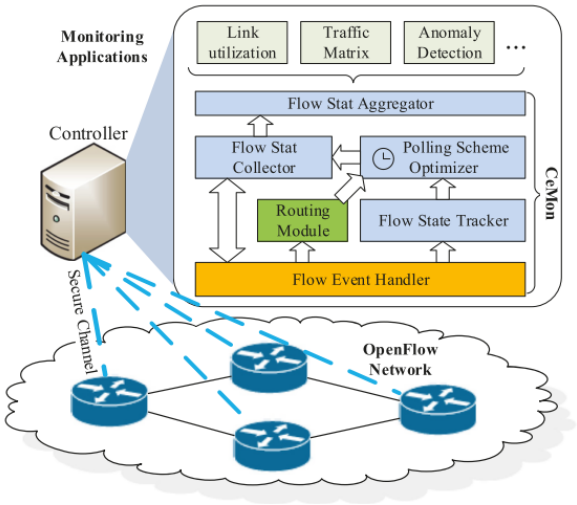


Figura 16 – CeMon - (SU et al., 2015)

O sistema desenvolvido (SU et al., 2015) apresentou resultados em que houve uma redução de 50% na sobrecarga provocada por tarefas de monitoramento, onde a queda na precisão para esse tipo de tarefa é desprezível.

O artigo (KWON et al., 2015) apresenta um protocolo denominado BASE para combater IP *spoofing*. A partir do IP *spoofing* é possível forjar o endereço IP de origem dos pacotes, possibilitando iniciar ataques anonimamente. BASE se trata de um protocolo incremental onde diferente de trabalhos anteriores, as suas três propriedades são satisfeitas. Na primeira, a implantação inicial do protocolo propicia benefícios para usuários. Enquanto na segunda, os benefícios aumentam de acordo com o prosseguimento da implantação (A medida em que o protocolo é implantado nos roteadores). Por último, o protocolo possui atributos que permitem a sua eficiência em uma implantação de pequena escala. Sendo assim, o BASE preenche essas três lacunas apresentando uma arquitetura focada para redes SDN. Nessa arquitetura são utilizadas três técnicas: *Message Authentication Code* (MAC) (GUO; CHIUEH, 2005), *One-way hash chains* (HU; JAKOBSSON; PERRIG, 2005), e *Packet marking* (BURCH; CHESWICK, 2000).

A primeira é responsável por confirmar que uma mensagem se originou de uma certa origem. Dessa forma, essa técnica é responsável por manter a autenticidade e integridade de um pacote. Enquanto a segunda possui a função de gerar valores de marcação para cada pacote, em que diversos valores são gerados para um mesmo dado, compondo uma cadeia. Já a terceira técnica consiste em marcar os pacotes com informações parciais sobre os seus caminhos conforme eles são encaminhados pelos roteadores. Essa última técnica é responsável pela marcação e filtragem dos pacotes. Isso posto, BASE executa como uma aplicação no controlador SDN da rede, permitindo que o aplicativo defina regras de marcação e filtragem, controlando o encaminhamento de pacotes.

Esse artigo (CHEN; YU, 2016) propõe uma arquitetura colaborativa de prevenção de intrusão denominado CIPA, cujo foco se trata de combater intrusões coordenadas, como por exemplo, DDoS, *Worms*, e *Botnets*. Ela é implantada como uma rede virtual de rede neural artificial. A partir da tecnologia de redes SDN é possível aproveitar a manipulação paralela dos neurônios a partir dos *switches* programáveis, onde cada um desses dispositivos virtualiza um ou mais neurônios. Consequentemente, CIPA pode dispersar o seu poder computacional e detectar ataques coordenados por meio de uma visão global. Isso posto, a abordagem de redes neurais artificiais consiste de neurônios que se comunicam por meio de ligações, com o intuito de tentar replicar o funcionamento do cérebro humano. Sendo assim, na arquitetura proposta são utilizadas um conjunto de redes neurais sobrepostas do tipo BP (*Back propagation*), além de um mecanismo de sincronização para lidar com atrasos na transmissão das mensagens entre os neurônios. Onde uma característica da rede BP se deve a atribuição de um peso a cada ligação, representando a força de conexão entre os neurônios. Sendo assim, a primeira camada da arquitetura desenvolvida se trata do *Input Layer*. Ela contém os neurônios de entrada, e é responsável por monitorar o tráfego de rede extraindo os seus atributos, traduzindo os seus formatos, e enviando os resultados preprocessados para a próxima camada. Na *Hidden Layer*, para cada atributo recebido do tráfego, são realizadas manipulações matemáticas simples, e os dados são enviados para a próxima camada, que pode se tratar de outra camada do tipo *Hidden Layer*, ou pode se tratar da *Output Layer*. No *Output Layer*, são recebidos os resultados da última camada, em seguida é realizada uma operação de mapeamento similar a camada anterior, e os dados são apresentados como os resultados da detecção. Por último, para implementar o *Mitigation Module*, mensagens de alerta são enviadas para uma parte ou todos os *switches* da rede, dessa forma, os *switches* podem localizar as fontes da intrusão, e construir regras de filtragem para o tráfego suspeito. O trabalho (CHEN; YU, 2016) proposto apresenta uma arquitetura escalável e de fácil aplicabilidade em redes de larga escala, além de apresentar bons resultados na detecção de tráfego suspeito. A figura 17 apresenta a arquitetura descrita.

Nesse artigo (SHANG et al., 2018) é proposta uma arquitetura onde um controlador distribuído é implementado baseado em um mecanismo de comunicação de multi-granularidade visando melhorar a segurança em um único domínio de controlador ou múltiplos domínios. Essa arquitetura é baseada no esquema de segurança SDN e consiste em dois módulos: módulo

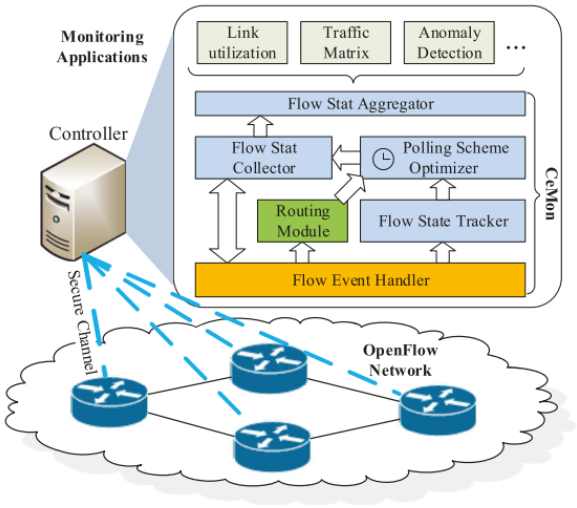


Figura 17 – CIPA - (CHEN; YU, 2016)

de controle básico, e módulo de customização de multi-granularidade. O módulo de multi-granularidade é composto por cinco componentes. O primeiro se trata do *Threat Defense*, responsável por detectar tráfego malicioso, e proteger o controlador. Além disso, ele é capaz de estabelecer regras de tráfego para que toda a rede possa executá-las. Já o componente *Flow Table Manager* procura combater conflitos entre tabelas de fluxo. No caso em que uma tabela de fluxo passa pela fase de detecção e nenhum conflito é detectado, ela é enviada para um determinado *switch* da rede. Enquanto para facilitar a recuperação rápida da falha de um controlador SDN, o componente *Backup* pode realizar *backup* e atualização de informações importantes referentes ao controlador, como políticas de segurança, e tabelas de fluxo. Por último, existe o *Application Manager* responsável por gerenciar a segurança dos controladores da rede, e o *Security Function Manager*, cuja função é de gerenciar os componentes citados anteriormente. Já o módulo de controle básico é responsável por realizar funções básicas de uma rede SDN, como gerenciar dispositivos, descobrir topologia, operações de leitura e escrita em uma tabela de fluxo, serviço de roteamento, e API para interface com o *data plane*.

4.2.4 NICE: Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems

De acordo com (CHUNG et al., 2013), o *framework* de detecção de intrusão NICE (*Network Intrusion Detection and Countermeasure Selection in Virtual Network Systems*) é proposto com o objetivo de impedir que ataques que exploram vulnerabilidades em um sistema de computação na nuvem comprometam as máquinas virtuais, evitando dessa forma ataques de larga escala como por exemplo DDoS (*Distributed Denial-of-Service*).

Para uma melhor detecção de ataques, o *framework* desenvolvido (CHUNG et al., 2013) incorpora procedimentos de análise para grafos de ataque no processo de detecção. Cada alerta gerado pode ser correlacionado de acordo com uma técnica proposta por (ROSCHKE; CHENG;

MEINEL, 2011). O sistema também possui contra-medidas baseadas na reconfiguração de redes virtuais através da abordagem de controle de rede conhecida como SDN (*Software Defined Networking*), em que as funções de uma rede podem ser programadas por meio de software *switches* e do protocolo OpenFlow. Dessa forma NICE se divide em duas etapas: Na primeira etapa um agente de detecção de intrusão em redes denominado NICE-A que se encontra presente em cada servidor da nuvem captura e monitora o tráfego gerado pelas máquinas virtuais. O agente periodicamente examina vulnerabilidades do sistema presentes em um servidor na nuvem para elaborar um grafo de cenário de ataque (*Scenario Attack Graph* - SAG). Esse grafo fornece informações sobre os caminhos que um atacante pode seguir. Por meio dele será decidido se uma determinada máquina virtual irá entrar em um estado de inspeção. Uma vez nesse estado, será aplicado sobre ela o método *Deep Packet Inspection* (DPI) ou inspeção profunda de pacotes, e/ou reconfigurações na rede para aumentar a probabilidade de detecção de ataques e a resiliência do sistema na presença de ataques que exploram as vulnerabilidades das máquinas virtuais.

A arquitetura do *framework* NICE (CHUNG et al., 2013) se divide em quatro componentes: O agente NICE-A presente em cada servidor físico (O agente se trata de um determinado IDS que será utilizado, no caso do trabalho em questão, o Snort), *Network Controller*, *VM Profile Server*, e *Attack Analyzer*, onde os três últimos estão localizados em uma central de controle conectado a *software switches* presentes em cada servidor da nuvem. De acordo com essa arquitetura, inicialmente a central de controle será notificada por um sistema de detecção de intrusão quando algum trafego anômalo for detectado. Em seguida o *attack analyzer* irá avaliar a severidade do alerta através do grafo de ataque e selecionar qual contra-medida executar, para inicia-la através do *network controller*. Uma vez que uma nova vulnerabilidade é descoberta ou uma contra-medida é iniciada, o grafo de ataque será atualizado. As contra-medidas são iniciadas pelo *network controller* através da reconfiguração virtual ou física dos *switches* da rede.

O sistema proposto (CHUNG et al., 2013) demonstra que por meio de sua utilização os riscos decorrentes da exploração e abuso de sistemas da nuvem por meio de ataques internos ou externos são reduzidos. No entanto o trabalho apenas investiga a abordagem de IDS de rede para ataques *zombie* exploratórios, no qual as vulnerabilidades do sistema são exploradas para que uma máquina se encontre sob o controle de um invasor. Por último, também é necessário investigar melhores soluções para IDS baseados em *host* para mitigar o problema de precisão da detecção de ataques.

4.2.5 SnortFlow: A OpenFlow-Based Intrusion Prevention System in Cloud Environment

Esse trabalho (XING et al., 2013) desenvolveu uma arquitetura para o ambiente da nuvem com o objetivo de suportar um IPS flexível e eficiente. Através dela, busca-se melhorar o *framework* NICE (CHUNG et al., 2013) através de um servidor de gerenciamento, cuja função se trata de coordenar alertas gerados pelo Snort em múltiplos servidores na nuvem. Por meio

dessa abordagem, a detecção de ataques multiníveis coordenados entre múltiplos servidores é facilitada. A arquitetura se baseia no protocolo OpenFlow, e na ferramenta de detecção Snort. O primeiro se trata de um protocolo de comunicação que permite a programação de redes SDN (MCKEOWN et al., 2008), enquanto o segundo é uma ferramenta de detecção de intrusão *open source* e gratuita.

A sua arquitetura se divide em quatro partes: *Cloud Cluster*, Controlador, Servidor SnortFlow, e OpenFlow *Switches*. A primeira parte é composta pelos servidores, onde em cada um deles se encontra um agente SnortFlow, responsável por monitorar a rede de máquinas virtuais, um switch virtual que conecta as VM, e a rede de VM monitorada pelo agente. Na segunda parte, alertas gerados por cada agente SnortFlow são enviados para o servidor Snortflow. Esse servidor é responsável por avaliar o status de toda a rede de servidores, e gerar ações ou tarefas que serão enviadas para o controlador SDN. O controlador, alimentado por ações geradas pelo servidor SnortFlow, irá injetar as tarefas de reconfiguração da rede nos OpenFlow *Switches*, para que elas possam ser executadas.

O IPS desenvolvido foi testado em um ambiente simples, em apenas um servidor, com o objetivo de testar a carga da rede, em duas configurações distintas. Além disso algoritmos de correlação de alertas e de reconfiguração da rede são necessários para que o IPS possa ser utilizado em ambientes mais complexos.

4.2.6 Security Analysis as Software-defined Security for SDN Environment

Esse trabalho propõe uma arquitetura que aumenta a segurança nas redes SDN. O sistema de detecção utilizado por ela é baseado em assinatura, e utiliza uma técnica de correlação de alertas (ROSCHKE; CHENG; MEINEL, 2011) por meio de grafo de ataque, visando melhorar a eficácia da detecção. A tecnologia SDN é utilizada para executar contra-medidas e mitigar ataques. No entanto, a técnica de correlação é melhorada por um método de clusterização no qual os alertas são agrupados de acordo com suas similaridades, enquanto a seleção de contra-medidas ocorre por meio da probabilidade condicional de cada vértice do grafo de ataque. Esse tipo de probabilidade considera a possibilidade de uma vulnerabilidade ser comprometida considerando todas as outras probabilidades presentes em cada vértice. Esses valores de probabilidade são derivados por meio do CVSS (MELL; SCARFONE; ROMANOSKY, 2006) referente a cada vulnerabilidade, e se referem a métrica de estado de segurança (MOUSSAID; TOUMANARI; AZHARI, 2017).

Esse artigo apresenta um modelo de segurança localizado na camada de aplicação de uma arquitetura SDN. Essa camada se encontra no topo de um controlador SDN e se comunica com ele por meio de uma interface denominada *North Bound Interface* (MOUSSAID; TOUMANARI; AZHARI, 2017).

O modelo proposto foi testado contra ataques de DDoS e ataques multiníveis em servidores da nuvem. As métricas utilizadas para medir seu desempenho foram as de consumo de CPU, porcentagem na redução de alarmes falsos, e estado de segurança de cada VM antes e depois de cada ataque realizado (MOUSSAID; TOUMANARI; AZHARI, 2017).

4.3 Comparação dos Trabalhos Relacionados

De acordo com (CHUNG et al., 2013) foi desenvolvida uma arquitetura de segurança na nuvem que utiliza conceitos de grafos de ataque e redes SDN para combater ataques que exploram vulnerabilidades no serviço de infraestrutura. No entanto, nesse modelo de segurança foi utilizada uma abordagem de detecção baseada em assinatura, o que limita a sua detecção a ataques no qual o sistema já possui conhecimento prévio. Outro problema se deve a escalabilidade, uma vez que a arquitetura precisa ser adaptada para nuvens de médio e grande porte. Já o trabalho de (XING et al., 2013) visa melhorar o *framework* NICE através da implementação de um módulo de correlação de alertas visando combater ataques coordenados de múltiplas etapas focado no ataque a diversos servidores. No entanto, novamente a arquitetura apresentada utiliza um sistema de detecção baseado em assinatura. Além disso, o módulo de correlação de alertas proposto, além de não ser implementado, não apresentava detalhes de como realizar a sua implementação, pois nenhuma técnica ou algoritmo é apresentado. O trabalho foi testado em um ambiente simples de um único servidor com um conjunto de máquinas virtuais. Outra arquitetura de segurança (MOUSSAID; TOUMANARI; AZHARI, 2017) busca melhorar a técnica de correlação de alerta do trabalho (ROSCHKE; CHENG; MEINEL, 2011), por meio de um método de clusterização. No entanto em comparação ao artigo (CHUNG et al., 2013), o seu algoritmo de seleção de contra-medidas precisa ser melhorado, pois não busca selecionar uma contra-medida ótima para cada cenário de ataque.

Diferente dos trabalhos anteriores, a abordagem proposta utiliza um sistema de detecção baseado em anomalia e dentro da abordagem imunológica. Esse sistema é utilizado em conjunto com uma técnica de correlação utilizada pelos artigos (CHUNG et al., 2013) (MOUSSAID; TOUMANARI; AZHARI, 2017). A primeira vantagem de se utilizar um IDS baseado em anomalia, se deve ao fato de que ataques não associados a nenhuma vulnerabilidade detectada por um grafo de ataque, ainda podem ser identificados e combatidos. Já outra vantagem, está relacionada a ataques de dia zero, em que uma nova ameaça associada a uma determinada vulnerabilidade pode surgir. No entanto um IDS baseado em assinatura apenas detecta ameaças para as quais já possui conhecimento prévio. Enquanto um sistema de detecção baseado em anomalia pode detectar e combater ataques de dia zero. Devido as vantagens mencionadas, a abordagem proposta possui uma maior cobertura de detecção, o que resulta em maiores chances de identificar ameaças em seus estágios iniciais, como ataques de reconhecimento. Outra vantagem da abordagem proposta se deve ao fato da análise do tráfego de rede ocorrer a nível de dados estatísticos, diferente dos trabalhos anteriores, onde a análise ocorre a nível de pacote. Isso

permite que o modelo de segurança proposto escale melhor para nuvens de médio a grande porte. Já para seleção de contra-medidas, seguindo a mesma metodologia de (CHUNG et al., 2013), será selecionada a contra-medida ótima para cada cenário de ataque. Contudo, a diferença em relação a (CHUNG et al., 2013), se deve ao modo de mapeamento de cada alerta gerado ocorrer apenas pelos atributos de origem e destino. Processo que será explicado em mais detalhes no capítulo seguinte.

De acordo com os resultados obtidos pelo artigo (MOUSSAID; TOUMANARI; AZHARI, 2017), a taxa de alertas falsos foi reduzida em 64% para ataques multinível e DDoS em sistemas de detecção baseados em assinatura. Já a abordagem proposta por esse trabalho conseguiu reduzir a taxa de alertas falsos para valores superiores a 90% para certos ataques na categoria DoS, alcançando em alguns casos uma redução de 100%, conforme será demonstrado nas seções de resultados.

Em relação a sistemas de detecção existe o trabalho de (IGBE; DARWISH; SAADAWI, 2016), no qual um mecanismo de detecção dentro da abordagem imunológica é desenvolvido para NIDS distribuídos. Uma vantagem dele se deve a utilização do algoritmo de seleção negativa em conjunto com algoritmo genético, com o objetivo de solucionar o problema da exaustão, na qual detectores válidos são gerados a um custo computacional maior. Outra vantagem está associada a troca de informações de intrusão entre os agentes do sistema, representados pelos NIDS.

Outros trabalhos dentro da abordagem imunológica buscam melhorar sua eficácia se utilizando de diferentes estratégias, como por exemplo, (SERESHT; AZMI, 2014). Em que o paradigma de rede imune é utilizado para colaboração entre os agentes, para que dessa forma auxilie na redução de alarmes falsos. De acordo com (CHEN; CHANG; WU, 2016), são utilizadas técnicas como PBIL, e CF, para melhorar problemas de classificação. Ainda existe o exemplo de (JHA; ACHARYA, 2016), que busca aperfeiçoar o processo de detecção por meio dos algoritmos *T-cell*, e *B-cell*. No entanto, a diferença da natureza da abordagem proposta para todos os outros trabalhos nessa área, se deve ao fato de considerar dados do ambiente no qual o sistema opera como forma de auxiliar no processo de detecção. Pois a técnica de correlação de alertas se utiliza de dados referentes ao meio, como a topologia da rede, e vulnerabilidades presentes em cada VM, para auxiliar no processo de detecção.

De acordo com os resultados obtidos por esse trabalho, a abordagem proposta apresentou melhor eficácia na detecção para todos as categorias de ataques na qual foi testado e comparado com o MAIS-IDS (SERESHT; AZMI, 2014). Isso ocorreu devido ao aumento na precisão da detecção causado por uma maior redução na taxa de alertas falsos. Esses resultados serão demonstrados na seção em que a eficácia dos sistemas são comparados para ataques nas classes DoS, R2L, U2R e *Probe*.

A seguir são apresentadas duas tabelas. Na tabela 3, sistemas de detecção são comparados em relação às técnicas, as bases de dados que cada um utiliza, e as métricas utilizadas (Acc

- Precisão, DR - Taxa de Detecção, FP - Falso Positivo, FN - Falso Negativo) por cada um deles para identificar em que atributo cada pesquisa procura melhorar em relação ao processo de detecção. Já a tabela 4 apresenta modelos de segurança para o ambiente da nuvem, onde mostra-se a abordagem de detecção utilizada por cada sistema, assim como as técnicas que cada um utiliza tanto na correlação de alertas quanto na prevenção a ataques. A abordagem proposta é comparada com os trabalhos citados em ambas as tabelas.

Tabela 3 – Comparativo entre Sistemas de Detecção

Artigos	***Técnicas de Detecção***	***Base de Dados***	***Métricas***
(ELHAJ; HAMRAWI; SULIMAN, 2013)	fuzzy logic	KDD-99	DR, FP
(SERESHT; AZMI, 2014)	NSA, CSA, INA	NSL-KDD	DR, FP, Acc
(IGBE; DARWISH; SAADAWI, 2016)	NSA, GA	NSL-KDD	DR, FP
(JHA; ACHARYA, 2016)	B-Cell, T-Cell	KDD-99	DR, FP
(AHMAD; IDRIS; KAMA, 2017)	DCA	DARPA99	DR, FP, FN
Sistema Proposto	NSA, CSA, INA, AGC	NSL-KDD	DR, FP, FN, Acc

Tabela 4 – Comparativo entre Modelos de Segurança

Artigos	***Detecção***	***Técnicas de Correlação***	***Prevenção***
(CHUNG et al., 2013)	Assinatura	AGC	Redes Bayesianas
(XING et al., 2013)	Assinatura	Não Definido	Não Definido
(MOUSSAID; TOUMANARI; AZHARI, 2017)	Assinatura	AGC	Redes Bayesianas
Sistema Proposto	Anomalia	AGC	Redes Bayesianas

5 Proposta

Este trabalho desenvolveu uma abordagem de detecção baseado em agentes dentro da abordagem imunológica para obter um sistema de detecção de intrusão distribuído, seguindo o modelo do MAIS-IDS (SERESHT; AZMI, 2014). Esse sistema funciona com uma técnica de correlação baseada em grafo de ataque, para reduzir a taxa de alarmes falsos. De acordo com a figura 18, a primeira parte refere-se ao processo de detecção. Nesse processo, cada máquina virtual possui uma instância do MAIS-IDS composta de agentes. Cada instância tem a função de analisar o tráfego de rede para detectar intrusões, ou de gerenciar a comunicação entre elas. Sendo assim, o processo de detecção em cada uma delas ocorre por meio de agentes detectores. Por outro lado, o gerenciamento da comunicação é realizado através do agente *orchestra*. A segunda parte do modelo proposto visa reduzir a quantidade de alarmes falsos por meio de uma técnica de correlação baseada em grafo de ataque. Para executar esta tarefa, cada alerta gerado por um agente detector vai para uma central de controle, onde existe um algoritmo de correlação (CHUNG et al., 2013), cuja a função é mapear e correlacionar alertas através de um grafo de ataque. Se o alerta for mapeado e correlacionado com sucesso, o ataque será confirmado e o grafo atualizado.

O sistema de detecção é implementado de acordo com (SERESHT; AZMI, 2014), e devido ao uso de agentes distribuídos, ele apresenta vantagens durante o processo de detecção. A primeira diz respeito à mobilidade, uma vez que os agentes podem migrar para outras máquinas virtuais. Este recurso ajuda a obter uma boa eficácia para um número reduzido de agentes, pois permite que eles realizem uma cobertura de toda a área de detecção. O segundo diz respeito à adaptação, pois permite que os detectores melhorem suas taxas de detecção. Essa melhora se deve à mudanças que a população de agentes detectores está sujeita. Consequentemente, a população repetidamente se renova enquanto o sistema está sendo executado. O terceiro diz respeito à autonomia, porque permite que os agentes se comportem sem a necessidade de um administrador. Através deste recurso um objetivo é atribuído com uma função de satisfação que recompensa os agentes quando o objetivo é alcançado. Desta forma, o comportamento dos agentes é orientado através de uma função de recompensa, melhorando a eficácia geral da detecção. Finalmente, coordenação, permitindo o trabalho coordenado entre eles, onde uma consequência deste se deve a redução na taxa de alarmes falsos, devido à cooperação dos detectores. Através da coordenação é possível sincronizar a comunicação entre os agentes, permitindo uma troca de informações entre eles, assim como um aprendizado através de uma função de recompensa.

O sistema proposto é composto principalmente por três componentes: agentes detectores, agentes antígenos, e agente *orchestra*. Agentes detectores são agentes móveis que são capazes de reconhecer agentes antígenos normais de agentes antígenos anômalos. Cada agente antígeno representa um padrão estatístico de rede, e eles podem ser classificados em *self* (dados normais)

e *non-self* (dados anômalos). Eles são agentes móveis que migram para outra máquina virtual se são identificados pelo melhor agente detector, aquele que possui a maior eficácia na detecção. A razão para a migração está associada a distância euclidiana, onde os antígenos identificados estão próximos do melhor detector. Assim sendo, eles estão mais propensos a obter uma alta eficácia na detecção, uma vez que se tornem detectores na máquina alvo. Já o agente *orchestra* se trata de uma entidade central que gerencia e coordena a comunicação entre outros agentes. Portanto, essa abordagem colaborativa distribui agentes entre máquinas virtuais, possibilitando a análise do tráfego de rede a nível de VM.

O IDS utiliza três algoritmos diferentes para melhorar a eficácia da detecção, que são seleção negativa, seleção clonal, e rede imune. Seleção negativa tem a função de remover agentes com baixa eficácia. Seleção clonal auxilia na criação e mutação de clones do melhor agente detector. Esses agentes clonados utilizam o algoritmo de rede imune para julgar antígenos que foram identificados como anômalos, e que se encontram armazenados na memória do melhor detector. Esse processo auxilia na confirmação de ameaças detectadas. Dessa forma, o algoritmo de rede imune auxilia o sistema a controlar, e reduzir o número de erros na identificação de dados anômalos.

Uma vez que um alerta é gerado pelo agente detector, ele precisa ser confirmado. Durante o processo de confirmação, os seus dados são analisados. Dessa forma, o alerta pode ser mapeado para o grafo de ataque por meio de sua origem e destino, ou seja, da máquina na qual a ameaça se originou para a máquina alvo.

O modelo de segurança proposto utiliza uma definição de grafo de ataque (CHUNG et al., 2013) com três tipos de nós. O primeiro se trata do nó de conjunção, onde é representado um ataque associado a uma determinada vulnerabilidade. O segundo é denominado nó de disjunção, cuja função é de representar o resultado de se explorar uma certa vulnerabilidade. Por último, existe o nó raiz para mostrar a etapa inicial de um cenário de ataque. Dentro do grafo, cada nó de conjunção representa um conjunto de *hosts* associados a endereços IP, vulnerabilidade, que está relacionada ao tipo de alerta, bem como alertas mapeados para ele (CHUNG et al., 2013). Sendo assim, para um IDS baseado em assinatura, quando um alerta é gerado, ele possui uma assinatura de ataque (um tipo de alerta) que pode estar associado a dados de vulnerabilidade. Esses dados podem ser encontrados em bancos de vulnerabilidades, como CVE (*Common Vulnerabilities and Exposures List*) (MELL; SCARFONE; ROMANOSKY, 2006), ou NVD (*National Vulnerability Database*) (National Institute of Standards and Technology, 2018), por exemplo. Consequentemente, o alerta pode ser mapeado para o grafo de ataque através dos seus dados de assinatura, origem, e destino. No entanto, uma vez que um IDS baseado em anomalia não utiliza nenhum tipo de assinatura durante o processo de detecção, é necessário usar outra forma de mapeamento. Nesse outro método, cada alerta é mapeado para um determinado vértice apenas de acordo com seus dados de origem e destino (ROSCHKE; CHENG; MEINEL, 2011). Isso permite a utilização de um sistema de detecção baseado em anomalia, em conjunto com

grafo de ataque, e correlação de alerta. Uma vez que informações da localização de um alerta podem ser extraídas de cada padrão estatístico do tráfego de rede analisado pelo IDS.

Uma vez que o alerta é mapeado para o grafo de ataque, ele precisa ser correlacionado com outros que foram gerados. Essa correlação ocorre por meio de um algoritmo (CHUNG et al., 2013), no qual a função é gerenciar um grafo de correlação de alertas (ACG), utilizando a abordagem em (ROSCHKE; CHENG; MEINEL, 2011). Este grafo é uma representação de diversos caminhos, onde cada um corresponde a um cenário de ataque, em que os alertas são organizados cronologicamente. Portanto, para cada um deles recentemente mapeado, há um vértice correspondente no grafo de ataque. Para este vértice, há vértices pais que podem ter alertas associados para os quais o mais recente pode se correlacionar. No entanto, outro fator que precisa ser analisado na correlação, se trata da diferença de tempo entre um novo alerta gerado e outro anterior. No caso dessa diferença de tempo exceder um certo limiar, não poderá haver uma correlação entre eles, pois eles pertencem a diferentes cenários de ataque. Sendo assim, conforme os alertas são gerados, os caminhos no ACG são formados. Dessa forma, se o processo de mapeamento e correlação forem bem sucedidos, o ataque é confirmado. A figura 18 apresenta a primeira parte do modelo de segurança proposto.

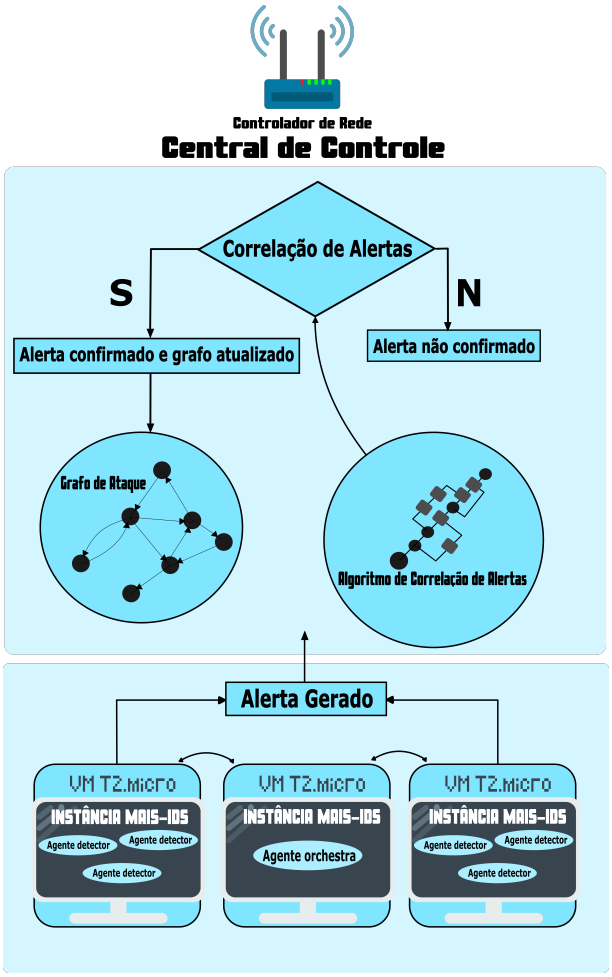


Figura 18 – Abordagem de Segurança - Parte 1

Na etapa de seleção de contra-medidas, cada vértice ou nó de conjunção do grafo de ataque possui uma vulnerabilidade associada a um valor de *base score* (BS), de acordo com o sistema CVSS. Esse valor corresponde a uma probabilidade que a vulnerabilidade possui de ser comprometida. Sendo assim, por meio do *base score*, presente em cada vértice, são derivadas probabilidades condicionais de acordo com (FRIGAULT; WANG, 2008). Uma probabilidade condicional se refere a probabilidade em que uma vulnerabilidade possui de ser comprometida, considerando todos os outros vértices do grafo. Para o cálculo dos valores condicionais, é utilizada uma técnica de redes Bayesianas (BN) para modelar o estado de segurança da rede. O grafo gerado pelo *software* MulVAL (OU; GOVINDAVAJHALA; APPEL, 2005), é modelado de acordo com um caso especial de BN. Sendo assim, quando a rede sofre um ataque, e um alerta gerado é mapeado e correlacionado com outros, o sistema irá selecionar uma contra-medida baseada na tecnologia de redes SDN. Essa seleção ocorre de acordo com o algoritmo apresentado em (CHUNG et al., 2013), por meio da análise de três valores: benefício, custo, e intrusividade. Benefício se refere à variação da probabilidade condicional no vértice alvo do alerta gerado, antes e depois do ataque realizado. O vértice alvo se refere ao vértice final do grafo, onde todos os caminhos convergem. Intrusividade, está relacionado ao efeito negativo que uma contra-medida SDN possui no nível de acordo de serviço (SLA - *Service Level Agreement*). Já o custo, se refere ao custo computacional que ela possui em termos de recursos utilizados para a sua execução. Esses três valores em conjunto compõem a métrica ROI (*Return of Investment*) (CHUNG et al., 2013). Dessa forma, será selecionada a contra-medida ótima para cada cenário de ataque, ou seja, a que possui o maior valor ROI. Uma vez que a rede SDN precisa ser reconfigurada para mitigar uma ameaça, um novo grafo de ataque precisa ser gerado, onde novos valores de probabilidade condicional para cada vértice será calculado. A figura 19 apresenta um diagrama de ações seguido pela abordagem proposta.

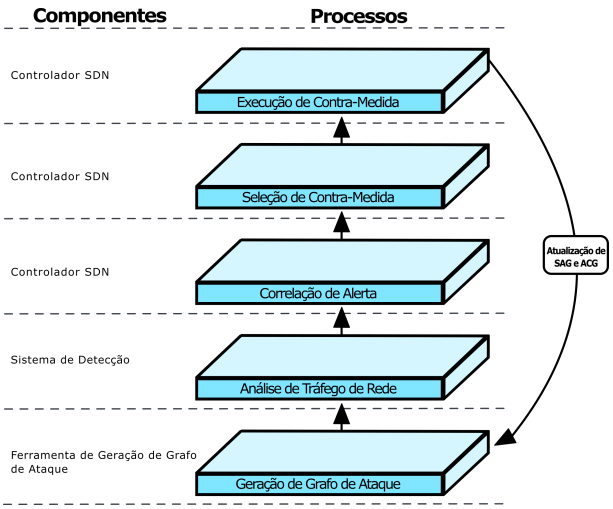


Figura 19 – Método Proposto

O sistema proposto pode auxiliar na detecção de diferentes classes de ataques, como por

exemplo, ataques de reconhecimento ou de negação de serviço, já que tais ameaças podem causar mudanças nos dados estatísticos do tráfego, sondando ou sobrecarregando uma rede. Além disso, a técnica de correlação utilizada pode auxiliar na confirmação de que uma certa vulnerabilidade está sendo realmente comprometida no momento, uma vez que essa técnica procura reduzir a taxa de alarmes falsos. Uma vez que o ataque é confirmado, ele poder ser mitigado através da execução de contra-medidas por meio da tecnologia SDN. Um exemplo seria em um ataque de *Arpspoofing*, que não seria possível em um cenário onde uma VM suspeita é isolada para outro *switch* virtual, por meio de uma reconfiguração na rede (KWON; AHN; CHUNG, 2004). A figura 20 representa todo o modelo de segurança proposto.

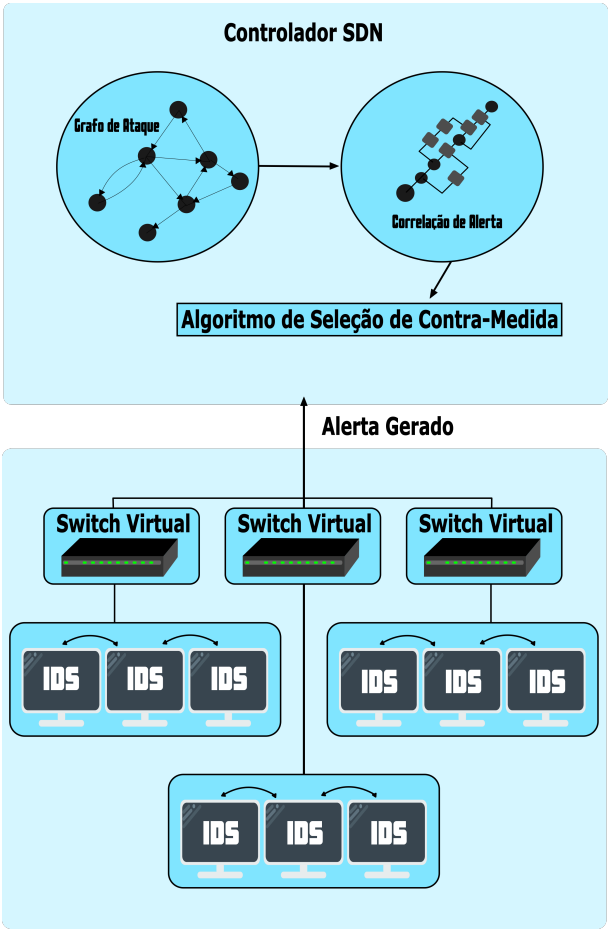


Figura 20 – Abordagem de Segurança - Parte 2

6 Cenários Experimentais

Esse capítulo descreve os cenários experimentais utilizados para avaliar a abordagem proposta. Primeiro são descritos os *softwares* utilizados para realização dos experimentos. Também são descritos os ambientes experimentais, assim como os ataques executados em cada cenário. Por último, é apresentada a base de dados utilizada para realização dos testes.

6.1 Softwares Utilizados

Para implementar a abordagem proposta, o serviço *Amazon Web Service* (AWS) (Amazon Web Service, 2018) foi utilizado para implementar o ambiente experimental. Para gerar um grafo de ataque referente a esse ambiente, primeiro foi utilizado o *scanner* Nessus (Tenable, 2018), para obter um arquivo com dados de vulnerabilidade da rede. Esses dados podem ser obtidos através de bases conhecidas (CVE, NVD) (The MITRE Corporation, 2018) (National Institute of Standards and Technology, 2018), que se tratam de bases *open source* e gratuitas. Em seguida a ferramenta MulVAL (OU; GOVINDAVAJHALA; APPEL, 2005) utilizou esse arquivo como entrada para gerar o grafo de ataque.

Todas as máquinas virtuais utilizaram o sistema operacional Linux. A partir desse sistema operacional, a captura do tráfego de rede foi realizada pela ferramenta *tcpdump*, onde essa captura forneceu os dados de entrada para gerar um *dataset* similar ao NSL-KDD (University of New Brunswick, 2018), de acordo com (PERONA, 2013). A partir desse *dataset* foi possível extrair todos os atributos do tráfego de rede necessários para a abordagem proposta.

Os algoritmos de Seleção Negativa, Seleção Clonal e de Rede Imunológica (BROWNLEE, 2011) foram implementados utilizando a linguagem de programação C. Em seguida eles foram adaptados para uma abordagem de agentes distribuídos (SERESHT; AZMI, 2014). Uma vez que os algoritmos se encontravam implementados, para a comunicação entre as diferentes instâncias do IDS, foram utilizados *sockets*. Através dos *sockets* as instâncias se comunicavam por meio do protocolo TCP. Já na abordagem original do sistema de detecção desenvolvido, conhecido como MAIS-IDS (SERESHT; AZMI, 2014), o protocolo de comunicação utilizado foi o Ejabberd. Uma das suas principais características se encontra relacionada à comunicação em tempo real (SERESHT; AZMI, 2014). No entanto, esse trabalho focou em testar o modelo de segurança proposto apenas em relação a sua eficácia na detecção, por essa razão não foi utilizado nenhum protocolo de mensagem instantânea.

Outro aspecto do sistema de detecção original, denominado MAIS-IDS, está associado a utilização de módulos de kernel para implementação dos agentes distribuídos. Enquanto na abordagem proposta, os agentes foram implementados por meio de variáveis dinâmicas. Cada

elemento de uma variável dinâmica correspondia a um determinado detector. Cada detector analisava os padrões estatísticos do tráfego de rede, e ao identificar algum padrão anômalo presente no *dataset*, armazenava-o dentro de sua memória. Essa memória corresponde a outra lista dinâmica dentro de cada elemento da lista de detectores. Sendo assim, as células de memória do melhor detector migravam para outra instância do sistema de detecção caso fosse confirmado como padrão anômalo por agentes de julgamento. Isso posto, o IDS se constituiu de agentes distribuídos, onde cada um deles correspondia a um determinado elemento de uma lista dinâmica. Consequentemente, os agentes detectores correspondiam à variável dinâmica de detectores, enquanto os agentes antígenos correspondiam à variável dinâmica de memória presente em cada elemento da lista de detectores. Já o agente *orchestra* se tratava de uma instância do sistema que gerenciava a comunicação entre outras instâncias compostas por agentes detectores, antígenos e de julgamento.

Os algoritmos de correlação e de seleção de contramedidas (CHUNG et al., 2013) também foram implementados na linguagem de programação C, visando facilitar a integração entre o sistema de detecção e a técnica de correlação de alertas. Para a implementação dessa técnica foram definidas variáveis dinâmicas correspondentes a grafos de ataque (SAG) e de correlação de alertas (ACG). As estruturas de dados definidas possibilitaram que o grafo de ataque gerado pelo *software* MulVAL fosse simplificado e convertido para outro modelo de grafo definido por (CHUNG et al., 2013). Esse modelo permitiu que os alertas gerados fossem mapeados para o grafo e correlacionados com outros previamente gerados. Permitiu também o cálculo das probabilidades condicionais correspondentes a cada vulnerabilidade, possibilitando a seleção de contramedidas SDN para mitigar ataques.

6.2 Ambiente Experimental

No primeiro ambiente experimental foram realizados ataques nas classes DoS, e *Probe*. Esse ambiente utilizou o serviço AWS (*Amazon Web Service*), e consistiu em uma rede pública com 8 instâncias de máquinas virtuais t2.micro (ECUs variáveis, 1 vCPUs, processamento de 2.5 GHz, família Intel Xeon, 1 GB de RAM, EBS apenas), cada uma contendo 8 GB de SSD (Disco de Estado Sólido). Este ambiente foi composto por um servidor DNS, um servidor de e-mail, e um servidor *web* Apache, no qual o MySQL foi utilizado para o banco de dados e a ferramenta *phpmyadmin* para administração do MySQL pela Internet. Totalizando na mesma rede pública 8 máquinas virtuais com o sistema operacional Ubuntu versão 16.04.4 LTS.

Para o segundo ambiente experimental, foram adicionadas mais três máquinas t2.micro com o mesmo sistema operacional das anteriores. Onde os serviços LamaHub (SOURCE..., 2018), Xplico (HOW..., 2018), DVWA (DAMN..., 2018) (*Damn Vulnerable Web Application*), e Telnet (TELNET..., 2018), foram introduzidos para realizar ataques nas categorias R2L, e U2R.

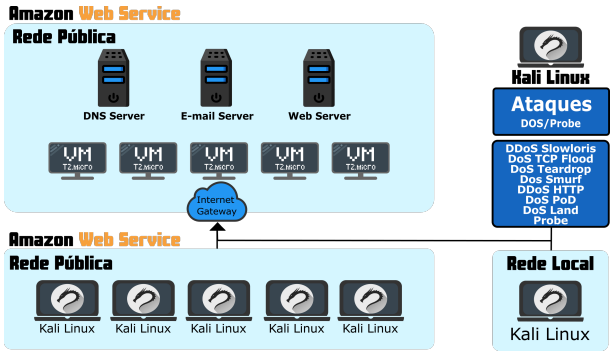


Figura 21 – Primeiro Cenário Experimental

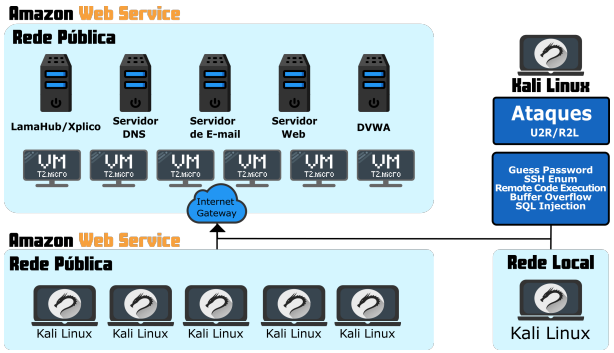


Figura 22 – Segundo Cenário Experimental

Para ataques nas classes DoS e *Probe*, foi utilizado o ambiente experimental apresentado pela figura 21, enquanto ataques nas classes R2L e U2R foram realizados no ambiente apresentado pela figura 22.

Os ataques foram realizados por máquinas virtuais t2.micro, e por um laptop com 8 GB de RAM, SSD de 240 GB (Disco de Estado Sólido), processador Intel Core2Duo com processamento de 2 GHz, onde ambos utilizaram o sistema operacional Kali Linux.

6.3 Ataques

No ambiente AWS foram realizados ataques em quatro categorias (DoS, U2R, R2L, *Probe*). Na categoria de negação de serviço, foram realizados ataques de DoS/DDoS. Entre eles, constam, **DoS Land**, onde o atacante envia pacotes TCP SYN com os mesmos IPs e portas de origem, e de destino. Isso faz a máquina alvo travar, enviando respostas para si mesma (Radware, 2018). O segundo, conhecido como **DoS Pod**, a máquina de destino é inundada com pacotes através do protocolo ICMP (Imperva, 2018). Outro tipo de negação de serviço implementado, simplesmente inunda a vítima com pacotes TCP. Além disso, dois ataques DDoS foram realizados através do protocolo HTTP na camada de aplicação. Um deles é conhecido como *Slowloris*, responsável por atacar *softwares* de servidores web como *Apache*. Essa ameaça estabelece múltiplas conexões para o servidor alvo através de requisições HTTP, e mantém elas

abertas o maior tempo possível. Subsequentemente, *Slowloris* envia cabeçalhos HTTP para cada requisição, resultando na queda do servidor *web* (Imperva, 2018). Já ataques de **DoS *Teardrop*** provocam sobreposição de fragmentos de pacotes IP, o que sobrecarrega servidores, e resulta na negação de serviço (Imperva, 2018). Também existem ameaças conhecidas como **DoS *Smurf***. Nesse tipo de ataque são realizadas uma série de solicitações ICMP para todos os *hosts* de uma rede por meio de um endereço de IP *broadcast*. Dessa forma, todos os *hosts* dessa rede respondem as solicitações enviadas para uma máquina vítima, resultando na sua sobrecarga e causando a negação de serviço (Radware, 2018).

Ataques de reconhecimento ou *probe* são executados por meio de varreduras na rede para identificar máquinas vulneráveis e comprometê-las. Ameaças desse tipo consistem em ações como varredura de IP, varredura de porta e escaneamento de porta. No primeiro, são realizadas varreduras na rede para identificar quais máquinas estão escutando (AMBEDKAR; BABU, 2015). No segundo, é realizada uma varredura para identificar se uma determinada porta se encontra aberta. Já no escaneamento de porta, uma única máquina é escaneada para identificar se uma determinada porta se encontra aberta.

Enquanto na categoria de R2L, foram realizados ataques que buscam obter usuário e senha de um determinado serviço, ou que buscam obter acesso não autorizado a uma máquina vítima como usuário local. Já na categoria U2R, foram realizados ataques de *buffer overflow* e *sql injection*. Ataques desse tipo buscam acesso não autorizado a uma máquina vítima como super usuário.

A tabela 5 apresenta as vulnerabilidades exploradas para cada classe de ataques. Na categoria DoS, a vulnerabilidade existente se encontra associada a ataques de negação de serviço conhecidos como *Slowloris*. Já na classe R2L a primeira vulnerabilidade está associada a obtenção de dados de usuário que utilizam a ferramenta OpenSSH (OpenBSD Project, 2018), enquanto a segunda está relacionada ao acesso não autorizado a uma determinada máquina.

Tabela 5 – Vulnerabilidades

Classes	**Vulnerabilidades**
DoS	CVE-2007-6750
U2R	-
R2L	CVE-2018-15473 CVE-2017-16666
Probe	-

Apesar da execução de diversos ataques presentes nas quatro classes, não são todos que se encontram associados a um "id"de vulnerabilidade que possa ser encontrado em bases conhecidas, como NVD (National Institute of Standards and Technology, 2018), ou CVE (The MITRE Corporation, 2018).

Para a classe DoS, a eficácia do sistema foi medida para cada ataque, enquanto para as classes U2R, R2L e *probe*, foi utilizado um único *dataset* contendo todos os ataques para cada fase experimental. A razão para isso se deve a uma menor quantidade de tráfego gerada por ataques U2R e R2L, enquanto ataques *probe* sempre ocorrem em conjunto e raramente de forma isolada.

Uma vez que os ataques foram executados, os detectores finais gerados pelo sistema de detecção foram testados sem a adição da técnica de correlação, e em conjunto com ela. Dessa forma foi possível medir a diferença na eficácia da detecção entre as duas versões da abordagem proposta.

6.4 Bases de Dados

Para testar o sistema proposto, dados estatísticos do tráfego de rede foram coletados. Como resultado, foram gerados dois tipos de dados para compor os *datasets*. O primeiro representa padrões normais, associados ao comportamento normal do sistema na ausência de ameaças. O segundo representa padrões anômalos, relacionados a ataques que podem ocorrer contra a rede. Portanto, com o objetivo de alcançar um melhor desempenho, apenas 19 dos mais importantes atributos de rede foram selecionados (*duration, service, flag, src_bytes, dst_bytes, wrong_fragment, urgent, hot, num_failed_logins, num_compromised, su_attempted, num_root, num_file_creations, num_shells, num_access_files, count, srv_count, dst_host_count, dst_host_srv_count*), de acordo com (FARID et al., 2009). No entanto, para o uso do sistema de detecção em conjunto com a técnica de correlação de alertas, foram utilizados dois atributos, além dos 19 citados anteriormente. Esses atributos foram os IP de origem e de destino de cada padrão estatístico da rede coletado. Como resultado, cada padrão de tráfego pôde ser mapeado para o grafo gerado pela ferramenta MulVAL (OU; GOVINDAVAJHALA; APPEL, 2005).

Os testes foram divididos em três fases: geração do detector, fase de treinamento, e fase de teste. A primeira fase compara cada detector gerado com os padrões normais de rede, a fim de remover cada detector que case com um deles (Seleção Negativa) no conjunto de dados normais. Na segunda fase, os agentes detectores aprendem e melhoram sua eficácia na detecção (Seleção Clonal e Rede Imune) por meio de um conjunto de treinamento. A última fase usa um conjunto de dados de teste para comparar a eficácia dos detectores finais na presença e ausência da técnica de correlação utilizada na abordagem proposta. Como resultado, cada avaliação de eficácia na detecção foi composta por um conjunto normal, outro de treino, e por último, um conjunto de teste. A figura 23 apresenta as etapas dos testes realizados para cada ataque. Primeiro são coletados tráfego normal e anômalo das VM do *Amazon Web Service*, para cada ataque efetuado. Em seguida, o sistema de detecção passa pelas etapas de geração de detectores, treino e teste.

Os conjuntos de dados foram gerados de acordo com o formato NSL-KDD (University of New Brunswick, 2018). O motivo para utilizar esse formato se deve a alguns problemas apontados

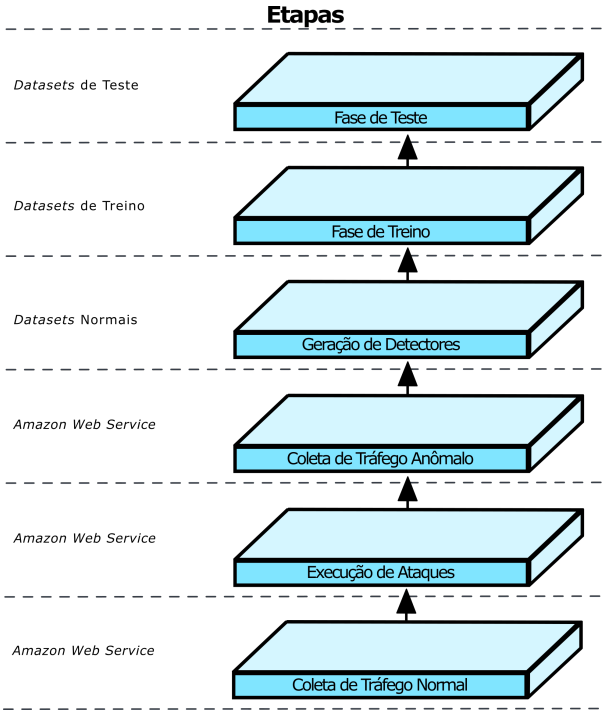


Figura 23 – Etapas dos Experimentos

pelos pesquisadores (GOGOI et al., 2012) (VASUDEVAN; HARSHINI; SELVAKUMAR, 2011) (MCHUGH, 2000) (TAVALLAEE et al., 2009) (MAHONEY; CHAN, 2003) em conjuntos de dados utilizados anteriormente. Esses problemas podem afetar a transparência da avaliação do IDS. Problemas como o valor de tempo de vida (TTL - *time-to-live*) dos pacotes. Os conjuntos de dados anteriores possuíam TTL com valores de 126 ou 253. No entanto, os pacotes do tráfego geralmente têm um TTL de 127 ou 254 (MCHUGH, 2000). Outro problema está relacionado à diferença entre os conjuntos de dados de treinamento, e teste. Isso se deve à adição de novos ataques para testar o conjunto de dados, o que leva a distorcer ou influenciar os métodos de classificação (MAHONEY; CHAN, 2003) (TAVALLAEE et al., 2009). O terceiro problema diz respeito a uma representação pobre de projeções de ataque *low footprint* (VASUDEVAN; HARSHINI; SELVAKUMAR, 2011).

De acordo com (TAVALLAEE et al., 2009), o NSL-KDD tenta resolver três problemas. Primeiro, ele tenta remover registros duplicados em conjuntos de dados de treinamento, e de teste de sua versão anterior, o KDDCUP99 (University of California, Irvine, 2018). Isso resulta na eliminação de sistemas de detecção tendenciosos a obter melhor eficácia para dados repetidos. Segundo, selecionar uma variedade de registros de diferentes partes do conjunto de dados original do KDD para obter resultados mais confiáveis dos sistemas de detecção. Terceiro, eliminando o problema de desequilíbrio entre o número de registros na fase de treinamento, e de teste para diminuir as taxas de falsos alarmes. O resultado de se atingir esses objetivos é um conjunto de dados mais confiável e imparcial para realização dos testes. As bases de dados geradas buscam solucionar os problemas enfrentados pelo KDDCUP99, seguindo o modelo do NSL-KDD. Sendo

assim, para a maioria dos cenários de ataque houve um equilíbrio entre a quantidade de dados de treino e de teste. Além disso, houve uma preocupação em eliminar dados redundantes dos *datasets* gerados.

7 Resultados Experimentais

Esse capítulo apresenta os resultados obtidos em relação a eficácia da detecção para a abordagem proposta, referente a cada cenário de teste. A eficácia do sistema foi testada individualmente e com correlação de alertas baseada em grafo de ataque. Para cada abordagem, os critérios utilizados para avaliar os resultados da detecção foram precisão (Acc), falso positivo (FP), falso negativo (FN), e taxa de detecção (DR). A eficácia da detecção foi avaliada para quantidades de detectores para o qual o sistema obteve melhores resultados. Para cada quantidade de detectores, o IDS foi executado 20 vezes para obter a média de cada métrica, assim como os valores de desvio padrão (STD), valores máximos (Max) e mínimos (Min), conforme (SERESHT; AZMI, 2014). A quantidade para que o IDS obtivesse a melhor eficácia, variou de acordo com cada cenário de ataque e de acordo com a quantidade de dados a serem analisados.

Nas seções seguintes as tabelas que são apresentadas na coluna a direita correspondem a eficácia dos detectores finais gerados, sem o auxílio da técnica de correlação, enquanto as da coluna a esquerda correspondem a eficácia deles com o auxílio da técnica em questão. Já cada imagem apresenta a eficácia do sistema referente às métricas (Acc, FP, FN, DR) para cada medida estatística utilizada (Média, STD, Max, Min).

A descrição dos resultados nas seguintes subseções são focadas nas alterações da eficácia ocorridas devido a adição da técnica de correlação ao sistema de detecção.

A tabela 6 exibe a quantidade de padrões estatísticos utilizados para cada ataque, dentro dos conjuntos de dados normais, de treino, e de testes.

Tabela 6 – Bases de Dados

Ataques	**Bases de Dados - Instâncias**		
	Normal	***Treino***	***Teste***
DDoS Slowloris	1322	2627	3438
DDoS HTTP	522	1035	2189
DoS Teardrop	1322	1259	759
DoS Smurf	1322	3958	4752
DoS Land	522	1059	926
DoS Pod	522	941	1123
DoS TCP	522	929	2389
Probe	1322	4329	4092
R2L	1322	4148	3623
U2R	1322	2633	2156

A tabela 7 apresenta a quantidade de dados anômalos e normais para as bases de treino e de teste, referente a cada ataque realizado.

Tabela 7 – Bases de Dados

Ataques	**Bases de Dados - Instâncias**			
	Treino		**Teste**	
	Normal	***Anomalia***	***Normal***	***Anomalia***
DDoS Slowloris	1958	669	1792	1646
DDoS HTTP	458	577	1589	600
DoS Teardrop	560	699	500	259
DoS Smurf	1958	2000	1792	2960
DoS Land	458	601	467	459
DoS Pod	458	483	467	656
DoS TCP	458	471	1589	800
Probe	1958	2371	1792	2300
R2L	1958	2190	1500	2123
U2R	1958	675	1500	656

7.1 Primeiro Ambiente Experimental

Nesse ambiente, a abordagem proposta foi comparada com o MAIS-IDS para as classes de ataque DoS e *probe*.

7.1.1 Negação de Serviço

Essa subseção reúne todos os ataques realizados na categoria de negação de serviço em um único *dataset*. A partir dela, os detectores finais gerados são comparados em relação a eficácia com o auxílio da técnica de correlação e em sua ausência. Para esse cenário as quantidades de detectores para o qual o sistema obteve maior eficácia correspondem a 50, 100, 150 e 200. A técnica de correlação melhorou a eficácia do sistema apenas para 50 detectores.

As tabelas a seguir apresentam os resultados obtidos, enquanto as figuras 24, 25, 26 e 27, apresentam os valores estatísticos para cada métrica analisada.

Tabela 8 – MAIS-IDS - DoS

50 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	83.309%	1.635%	19.929%	80.071%
STD	1.062%	7.127%	1.751%	1.751%
Max	84.054%	32.702%	22.974%	85.404%
Min	81.093%	0.0%	14.596%	77.026%

Tabela 9 – Abordagem Proposta - DoS

50 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	83.326%	1.534%	19.93%	80.070%
STD	1.046%	6.687%	1.749%	1.749%
Max	84.054%	30.682%	22.974%	85.39%
Min	81.093%	0.0%	14.61%	77.026%

De acordo com os resultados, é possível observar que para a quantidade de 50 detectores houve uma leve melhora na média da precisão. Isso se deve a redução de alarmes falsos devido a adição da técnica de correlação, resultando em um aumento da precisão. No entanto os valores máximo e mínimo da precisão permaneceram constantes, apesar da redução do seu desvio padrão. Considerando a métrica de falso positivo, houve uma pequena redução em sua média e desvio

Tabela 10 – MAIS-IDS - DoS

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	83.458%	0.0%	20.099%	79.901%
STD	1.057%	0.0%	1.284%	1.284%
Max	85.183%	0.0%	22.2%	81.996%
Min	81.73%	0.0%	18.004%	77.8%

Tabela 11 – Abordagem Proposta - DoS

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	83.458%	0.0%	20.099%	79.901%
STD	1.057%	0.0%	1.284%	1.284%
Max	85.183%	0.0%	22.2%	81.996%
Min	81.73%	0.0%	18.004%	77.8%

Tabela 12 – MAIS-IDS - DoS

150 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	83.804%	0.0%	19.68%	80.32%
STD	0.692%	0.0%	0.841%	0.841%
Max	84.076%	0.0%	22.2%	80.652%
Min	81.73%	0.0%	19.348%	77.8%

Tabela 13 – Abordagem Proposta - DoS

150 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	83.804%	0.0%	19.68%	80.32%
STD	0.692%	0.0%	0.841%	0.841%
Max	84.076%	0.0%	22.2%	80.652%
Min	81.73%	0.0%	19.348%	77.8%

Tabela 14 – MAIS-IDS - DoS

200 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	83.917%	0.0%	19.542%	80.458%
STD	0.502%	0.0%	0.61%	0.61%
Max	84.076%	0.0%	22.2%	80.652%
Min	81.73%	0.0%	19.348%	77.8%

Tabela 15 – Abordagem Proposta - DoS

200 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	83.917%	0.0%	19.542%	80.458%
STD	0.502%	0.0%	0.61%	0.61%
Max	84.076%	0.0%	22.2%	80.652%
Min	81.73%	0.0%	19.348%	77.8%

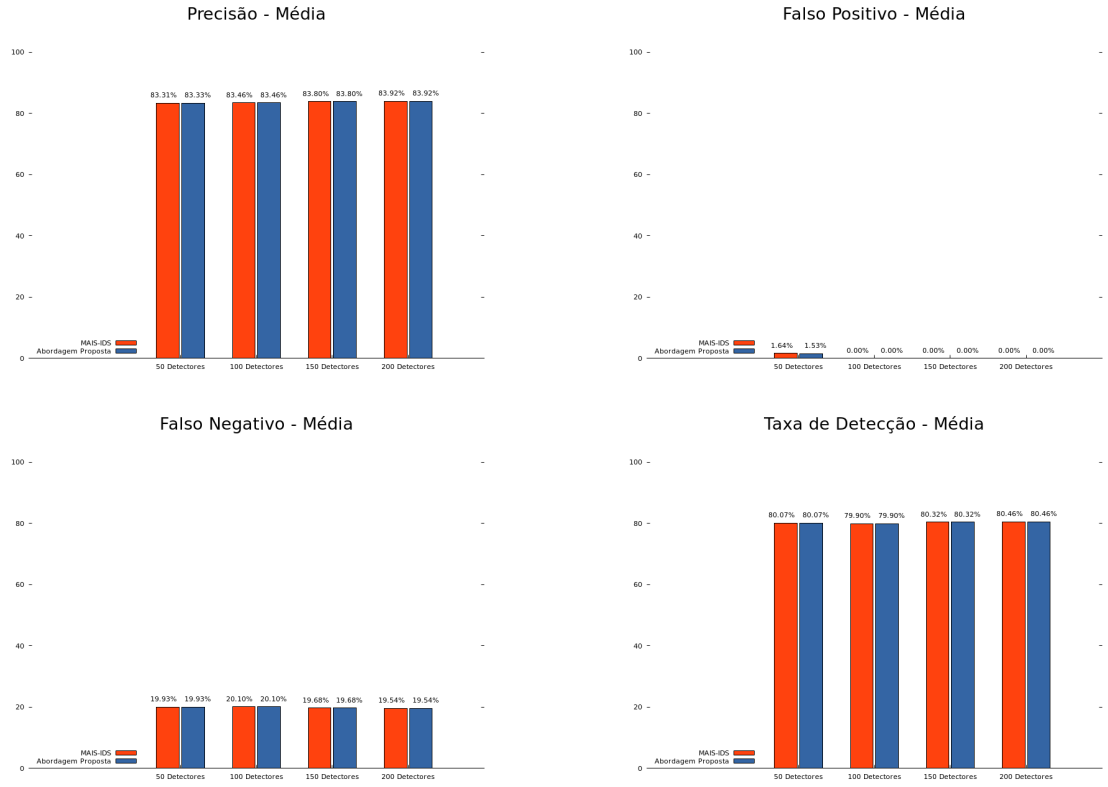


Figura 24 – DoS - Média

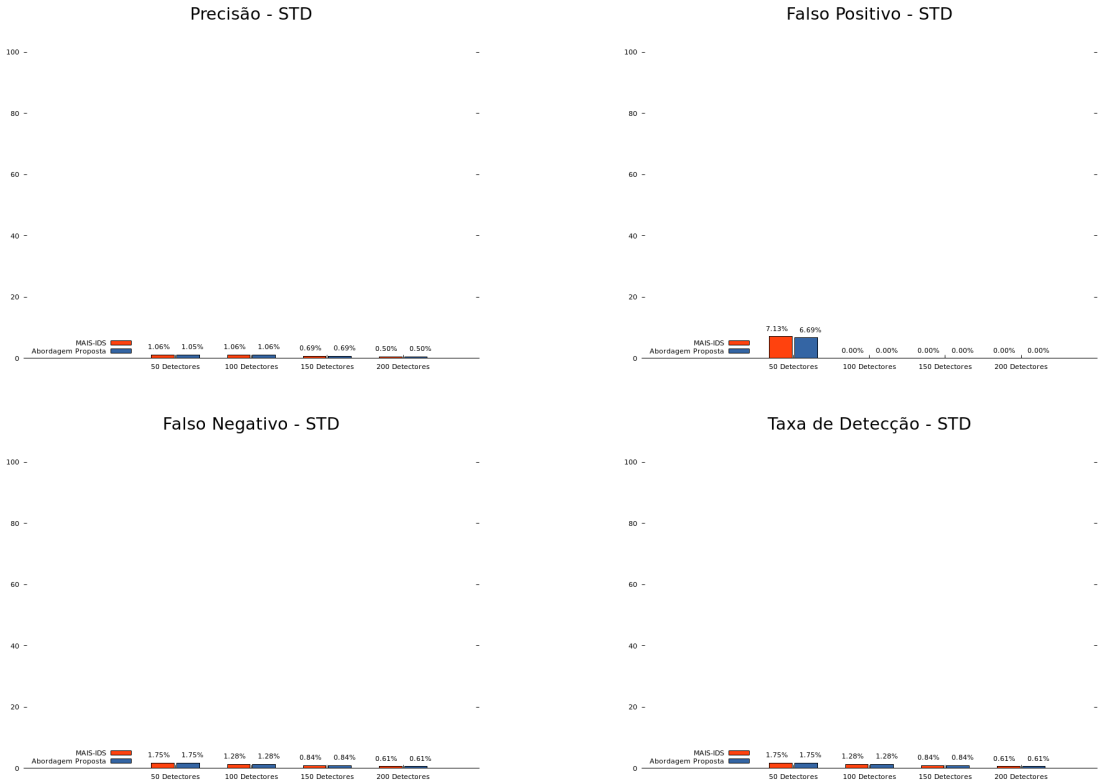


Figura 25 – DoS - STD

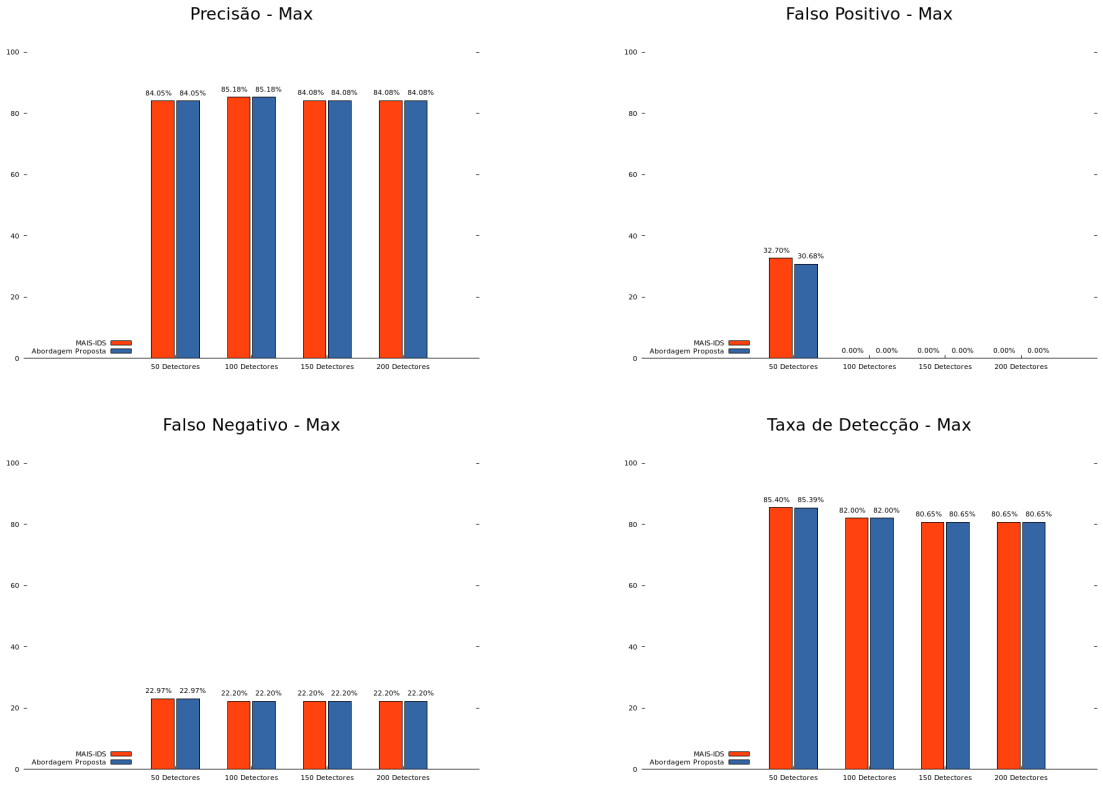


Figura 26 – DoS - Max

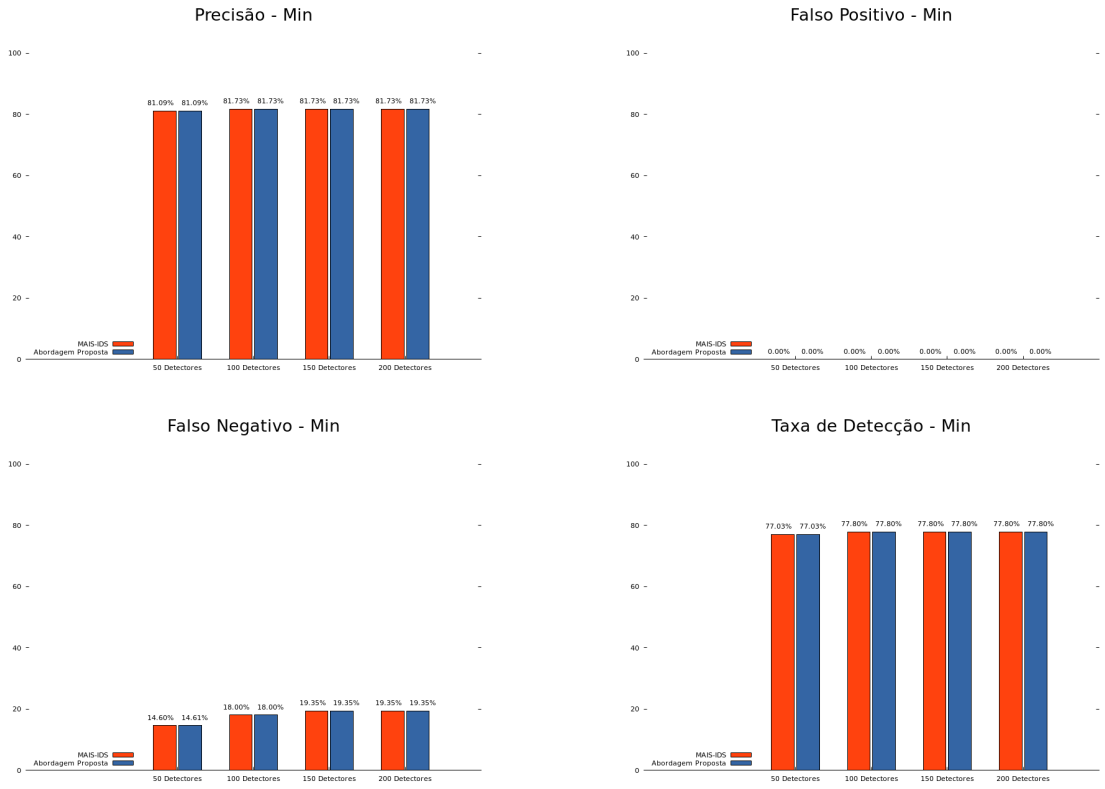


Figura 27 – DoS - Min

padrão. Além disso o seu valor máximo atingido também reduziu, enquanto o valor mínimo continuou em 0.0%, o que contribuiu na diminuição do STD. A média da taxa de falso negativo sofreu um pequeno aumento, enquanto o seu valor mínimo atingido também sofre um leve crescimento. Já em relação a taxa de detecção houve uma desprezível redução em sua média e desvio padrão. Porém, o valor mínimo alcançado permaneceu inalterado, enquanto o máximo decaiu levemente.

Para quantidades maiores de detectores a técnica de correlação não melhorou a eficácia da detecção. A razão para isso se deve a taxa de alertas falsos que permaneceu sempre em 0.0%. Contudo, pôde-se perceber que conforme a quantidade de detectores aumentava, a média da precisão também sofria incremento, alcançando o seu maior valor para 200 detectores.

7.1.1.1 DoS Land

Em relação a ataques de DoS Land o sistema apresentou sua melhor eficácia para as quantidades de 15, 20, 25 e 40 detectores. No entanto, a técnica de correlação apenas melhorou a eficácia do sistema para 20 detectores. Além disso, a métrica de falso negativo não sofreu alterações devido a sua adição.

As tabelas a seguir apresentam os resultados para as quantidades correspondentes, com o uso da técnica de correlação e sem a sua utilização. Em seguida as figuras 28, 29, 30 e 31

apresentam os resultados de acordo com medidas estatísticas para cada métrica.

Tabela 16 – MAIS-IDS - Land

15 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	81.744%	0.0%	36.83%	63.17%
STD	3.789%	0.0%	7.645%	7.645%
Max	82.613%	0.0%	70.153%	64.924%
Min	65.227%	0.0%	35.076%	29.847%

Tabela 17 – Abordagem Proposta - Land

15 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	81.744%	0.0%	36.83%	63.17%
STD	3.789%	0.0%	7.645%	7.645%
Max	82.613%	0.0%	70.153%	64.924%
Min	65.227%	0.0%	35.076%	29.847%

Tabela 18 – MAIS-IDS - Land

20 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	82.041%	1.0%	34.739%	65.261%
STD	3.141%	6.394%	10.075%	10.075%
Max	85.205%	29.336%	63.399%	100%
Min	68.575%	0.0%	0.0%	36.601%

Tabela 19 – Abordagem Proposta - Land

20 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	82.765%	0.032%	34.739%	65.261%
STD	4.938%	0.14%	10.075%	10.075%
Max	99.676%	0.642%	63.399%	100%
Min	68.575%	0.0%	0.0%	36.601%

Tabela 20 – MAIS-IDS - Land

25 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	82.613%	0.0%	35.076%	64.924%
STD	0.0%	0.0%	0.0%	0.0%
Max	82.613%	0.0%	35.076%	64.924%
Min	82.613%	0.0%	35.076%	64.924%

Tabela 21 – Abordagem Proposta - Land

25 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	82.613%	0.0%	35.076%	64.924%
STD	0.0%	0.0%	0.0%	0.0%
Max	82.613%	0.0%	35.076%	64.924%
Min	82.613%	0.0%	35.076%	64.924%

Tabela 22 – MAIS-IDS - Land

40 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	82.613%	0.0%	35.076%	64.924%
STD	0.0%	0.0%	0.0%	0.0%
Max	82.613%	0.0%	35.076%	64.924%
Min	82.613%	0.0%	35.076%	64.924%

Tabela 23 – Abordagem Proposta - Land

40 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	82.613%	0.0%	35.076%	64.924%
STD	0.0%	0.0%	0.0%	0.0%
Max	82.613%	0.0%	35.076%	64.924%
Min	82.613%	0.0%	35.076%	64.924%

Para 20 detectores houve uma pequena melhora na eficácia do sistema devido a adição da técnica de correlação. Para a métrica de precisão, a média sofreu um leve aumento, seguido também de um crescimento do STD. O valor máximo aumentou de 85.205% para 99.676%, enquanto o mínimo permaneceu constante, o que contribuiu para o aumento no desvio padrão. Já para a taxa de alertas falsos, a técnica de correlação reduziu a sua média em 96.8%. Isso resultou em uma diferença bem menor entre os valores máximo e mínimo alcançados, o que contribuiu para uma queda significativa no desvio padrão.

Para outras quantidades é possível observar que a média de alarmes falsos se encontra em 0.0%, por essa razão a adição da técnica de correlação não interferiu na eficácia. Outro ponto a ser observado se deve ao fato de que acima de 20 detectores o desvio padrão para todas as métricas permanece em zero. Isso significa que não houve desvio em relação a média para nenhuma delas, resultando nos mesmos valores máximo e mínimo para todas elas. No entanto, conforme a quantidade de detectores aumenta, ocorre uma redução no valor de falso negativo,

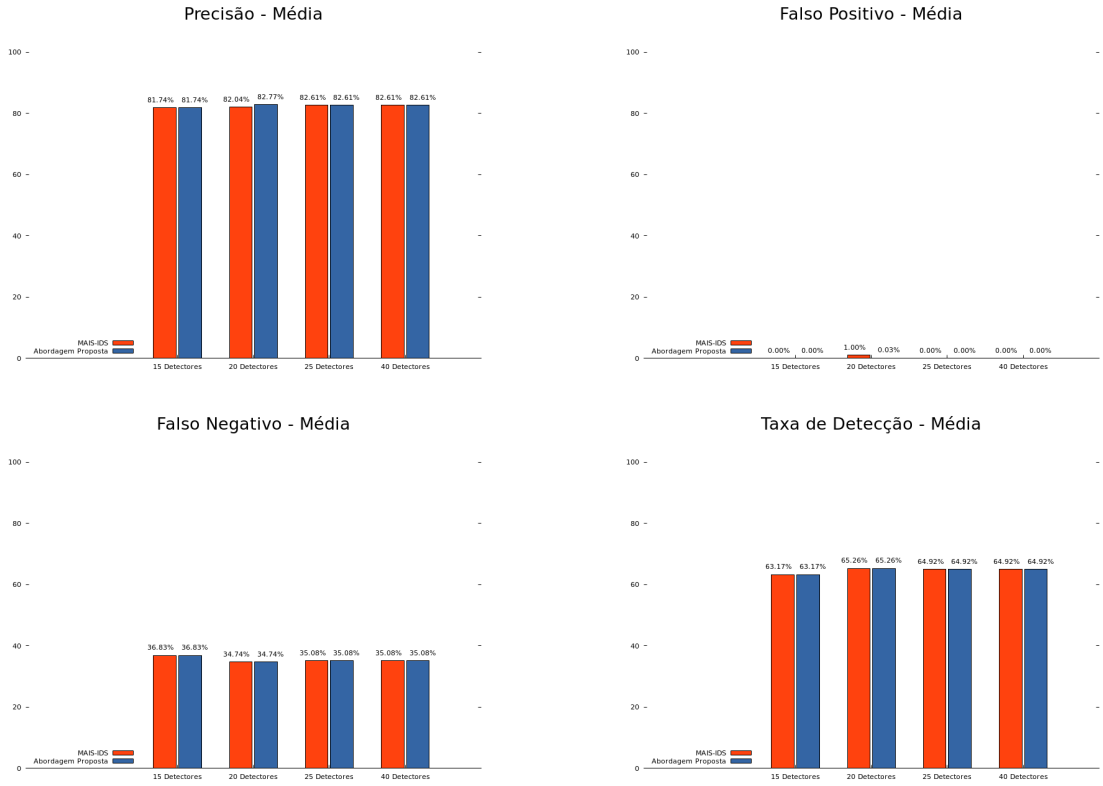


Figura 28 – DoS Land - Média

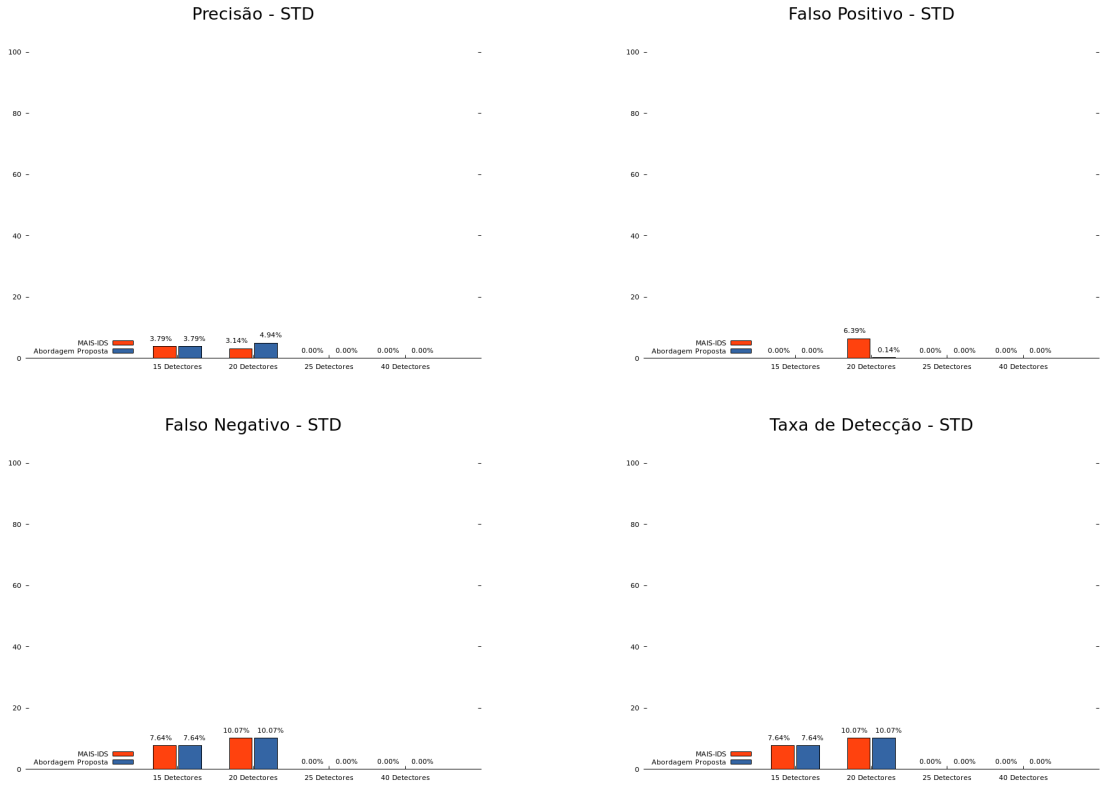


Figura 29 – DoS Land - STD

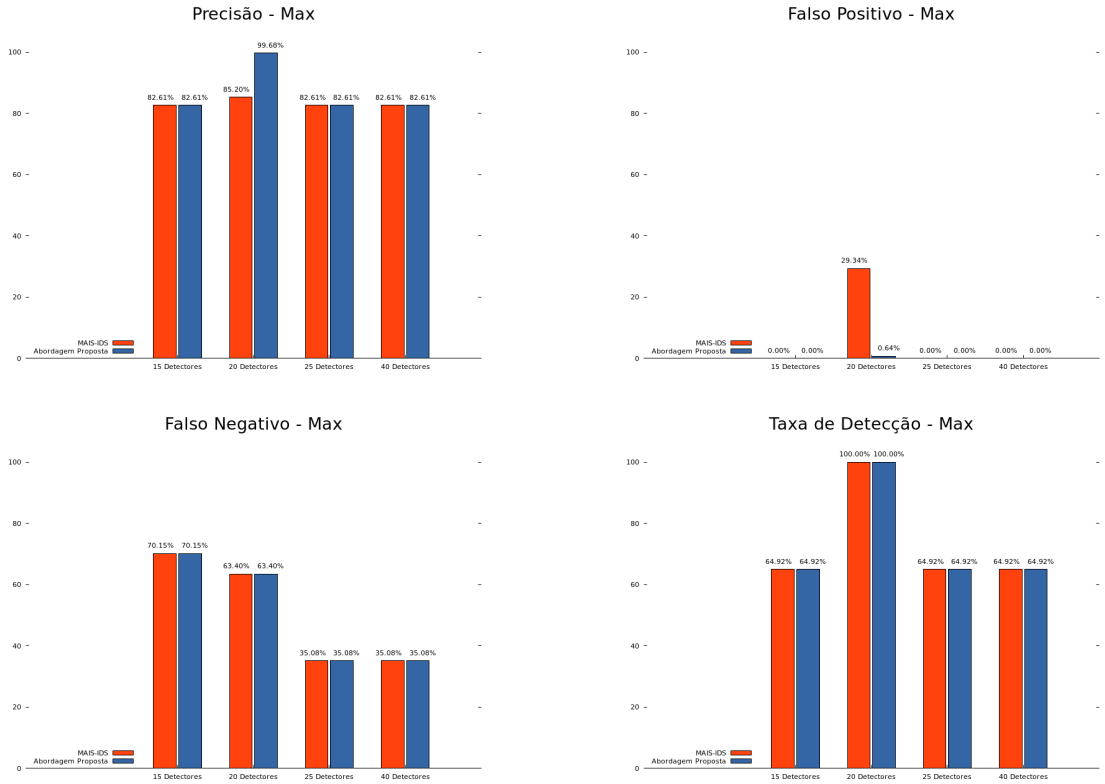


Figura 30 – DoS Land - Max

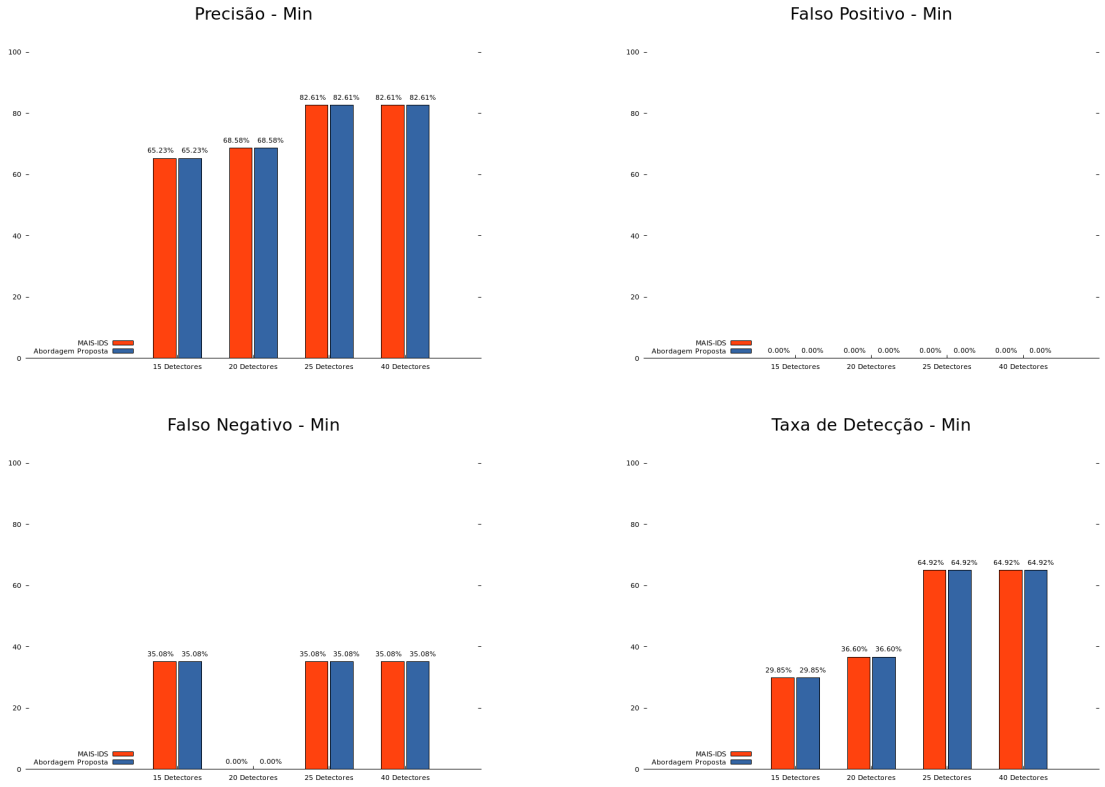


Figura 31 – DoS Land - Min

e aumento na taxa de detecção. Porém a partir de 25 detectores, ocorre um pequeno aumento na quantidade de falso negativo, resultando em uma menor taxa de detecção, além disso essas métricas se estabilizam e não apresentam alterações em seus valores.

7.1.1.2 DoS PoD

O ataque de negação de serviço DoS PoD apresentou melhor eficácia para 80, 100, 150 e 200 detectores. A técnica de correlação melhorou a eficácia para todas as quantidades testadas.

As tabelas a seguir apresentam os resultados da eficácia em conjunto com as figuras 32, 33, 34 e 35.

Tabela 24 – MAIS-IDS - PoD

80 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	65.557%	35.76%	33.506%	66.494%
STD	16.945%	8.145%	23.441%	23.441%
Max	86.554%	48.608%	63.415%	94.665%
Min	46.661%	24.625%	5.335%	36.585%

Tabela 25 – Abordagem Proposta - PoD

80 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	80.16%	0.642%	33.506%	66.494%
STD	13.406%	8.145%	23.441%	23.441%
Max	96.171%	1.927%	63.415%	94.665%
Min	62.956%	0.0%	5.335%	36.585%

Tabela 26 – MAIS-IDS - PoD

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	69.862%	33.48%	27.759%	72.241%
STD	16.173%	10.913%	23.811%	23.811%
Max	87.533%	48.394%	61.738%	96.037%
Min	45.236%	4.925%	3.963%	38.262%

Tabela 27 – Abordagem Proposta - PoD

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	83.272%	1.231%	27.759%	72.241%
STD	13.7%	0.885%	23.811%	23.811%
Max	96.972%	2.57%	61.738%	96.037%
Min	62.867%	0.0%	3.963%	38.262%

Tabela 28 – MAIS-IDS - PoD

150 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	78.972%	34.411%	11.502%	88.498%
STD	10.112%	6.715%	13.348%	13.348%
Max	85.396%	50.535%	51.829%	95.122%
Min	49.154%	27.623%	4.878%	48.171%

Tabela 29 – Abordagem Proposta - PoD

150 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	92.614%	1.606%	11.502%	88.498%
STD	7.581%	0.641%	13.348%	13.348%
Max	96.705%	2.57%	51.829%	95.122%
Min	69.724%	0.0%	4.878%	48.171%

Tabela 30 – MAIS-IDS - PoD

200 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	80.98%	33.812%	8.491%	91.509%
STD	5.892%	5.271%	7.394%	7.394%
Max	84.951%	47.109%	40.244%	95.732%
Min	56.901%	26.981%	4.268%	59.756%

Tabela 31 – Abordagem Proposta - PoD

200 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	94.252%	1.895%	8.491%	91.509%
STD	4.385%	0.539%	7.394%	7.394%
Max	96.349%	2.784%	40.244%	95.732%
Min	75.423%	1.071%	4.268%	59.756%

Para a quantidade de 80 detectores assim como todas as outras, não houve variação nas taxas de detecção e falso negativo. Sendo assim, para 80 detectores o sistema apresentou melhorias na média, valores mínimos e máximos, para as métricas de precisão e falso positivo, onde a redução na média de alertas falsos foi de 98.2%. Em relação ao desvio padrão o sistema diminuiu os seus valores para a métrica de precisão, enquanto para as taxas de detecção, falso

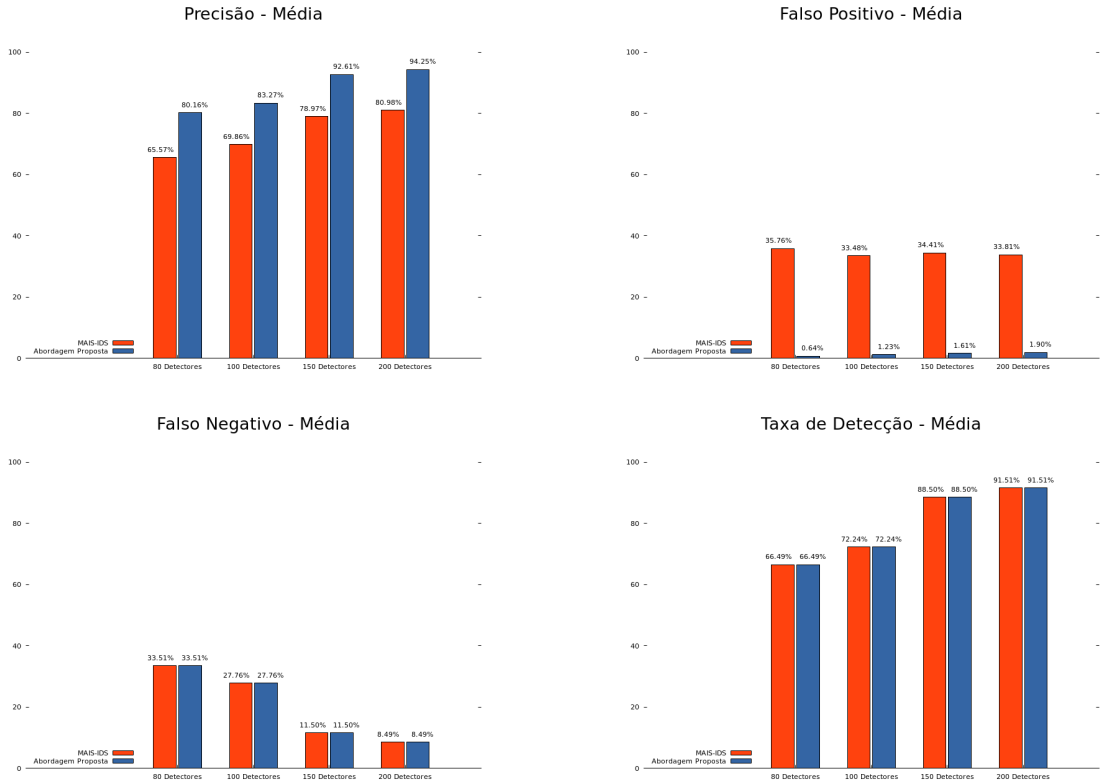


Figura 32 – DoS PoD - Média

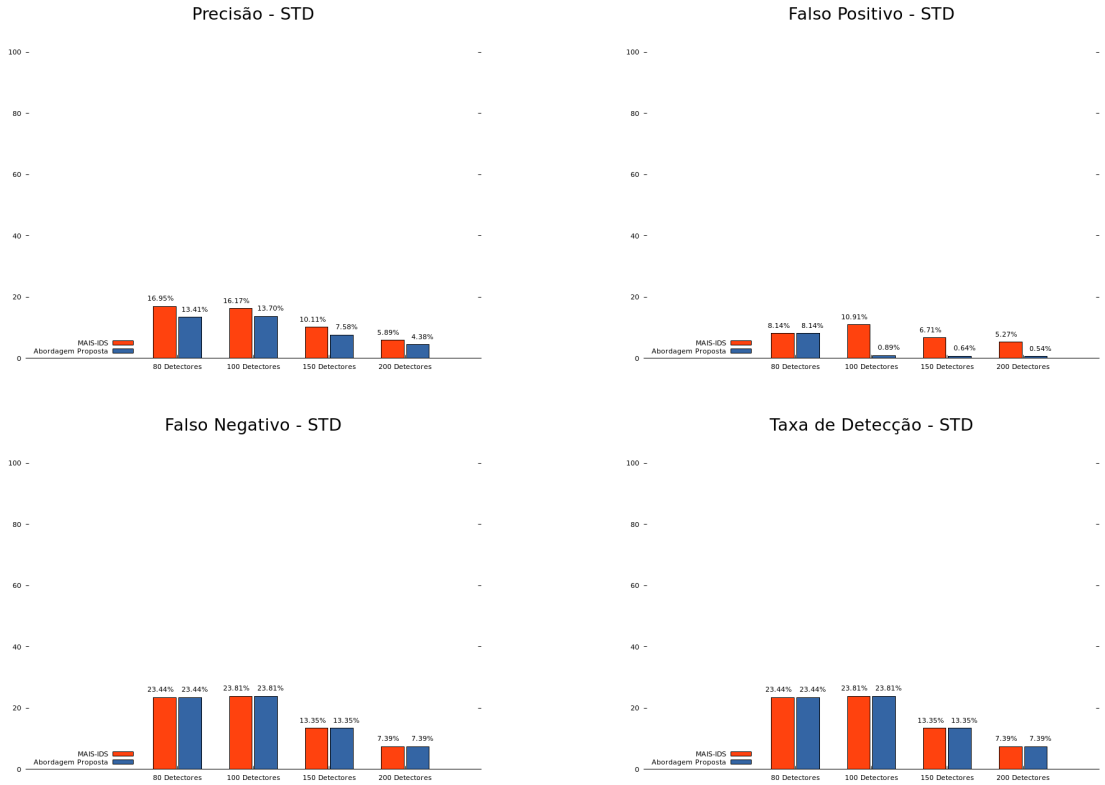


Figura 33 – DoS PoD - STD

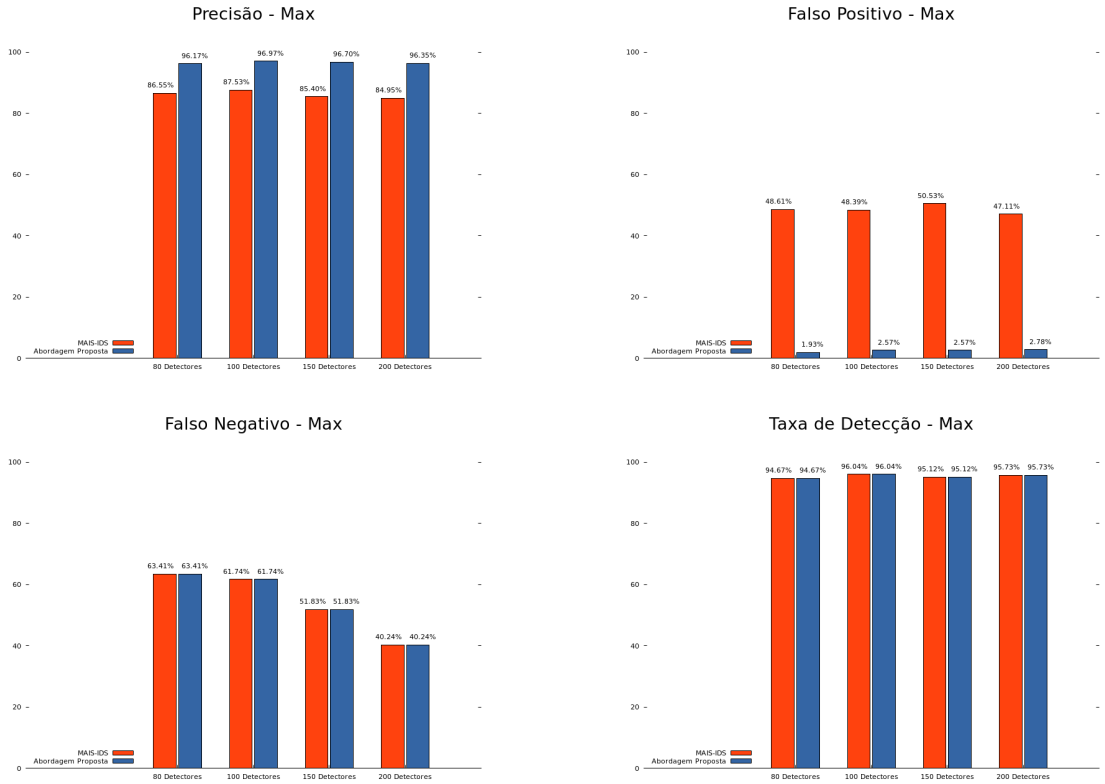


Figura 34 – DoS PoD - Max

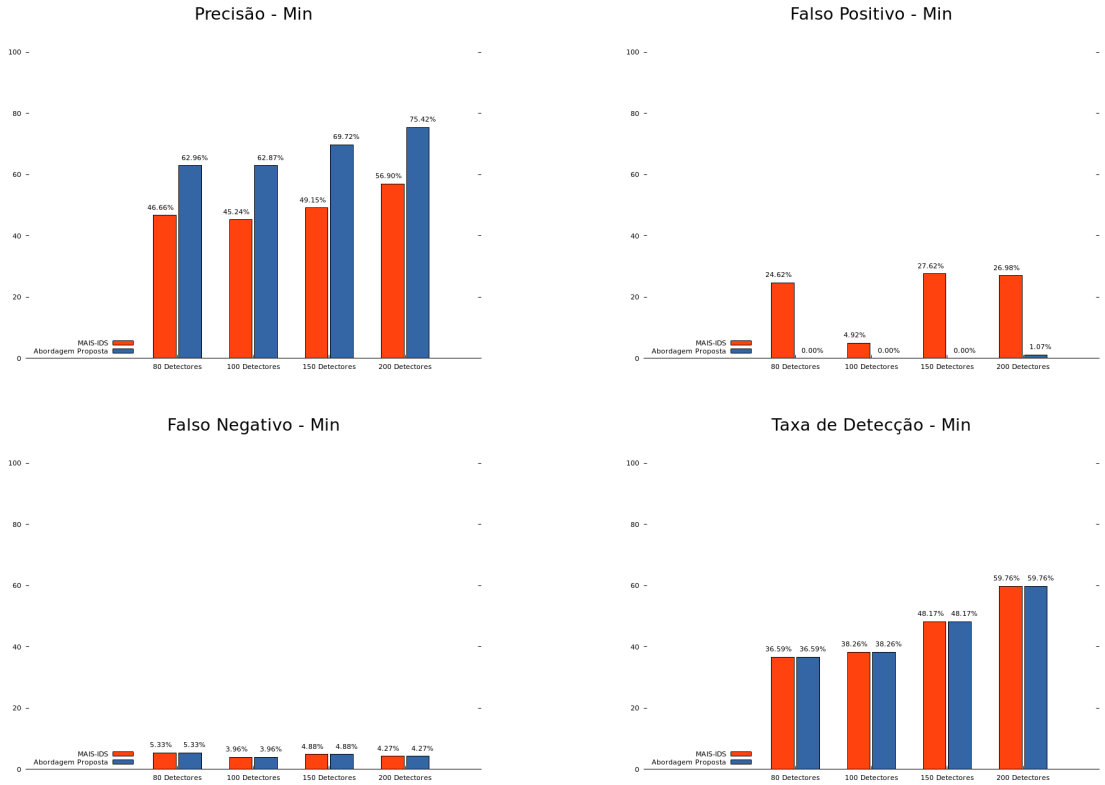


Figura 35 – DoS PoD - Min

positivo e falso negativo, o desvio padrão permaneceu inalterado. No decorrer de 20 execuções a métrica de precisão obteve uma estabilidade maior devido a redução de seu STD.

A quantidade de 100 detectores apresentou uma grande diferença na média das métricas de precisão e falso positivo. Em relação a média, a precisão obteve um aumento considerável, enquanto o valor de alertas falsos diminuiu em 96.3%. O desvio padrão reduziu para a precisão e falso positivo, enquanto para as taxas de detecção e falso negativo, permaneceu inalterado. Também houve uma grande variação nos valores máximos alcançados. A precisão máxima atingida com o auxilio da técnica de correlação alcançou 96.972%, contra 87.533% na ausência dela. Já os valores máximos de alertas falsos alcançaram 48.394% contra 2.57% com correlação. Houve grande diferença também em relação a precisão mínima atingida, com o auxílio do método o sistema alcançou valores mínimos de 62.867% contra 45.236%. O método utilizado atingiu uma redução de 100% em alertas falsos, enquanto o valor mínimo sem a utilização de correlação alcançou 4.925%.

Para 150 detectores, em relação a média das métricas, houve um considerável aumento na precisão, variando de 78.972% para 92.614%, enquanto houve uma redução de 95.3% na quantidade de alertas falsos. O desvio padrão sofreu uma considerável redução nas métricas de precisão e falso positivo, onde a última sofreu uma maior redução, o que mostra que a taxa de alertas falsos permaneceu sempre reduzida sem sofrer grandes variações em relação a média. Em relação aos valores máximos atingidos, a adição da técnica aumentou a precisão e reduziu o valor de falso positivo, consideravelmente. Enquanto para valores mínimos, houve melhorias nas métricas de precisão e falso positivo, onde a adição de correlação conseguiu reduzir a taxa de falso positivo em 100% em alguns casos.

A quantidade de 200 detectores apresentou resultados onde em relação a média o sistema apresentou melhorias significativas nas métricas de precisão e falso positivo, onde o segundo foi reduzido em 94.4%. Já para o desvio padrão, ocorreu uma redução nas duas primeiras métricas. Consequentemente, os valores de precisão e falso positivo permaneceram mais estáveis no decorrer de 20 execuções. Para valores máximos e mínimos houve melhoras significativas para a precisão e falso positivo.

Os resultados referentes a esse ataque apresentaram uma média superior para alarmes falsos, resultando em um maior ganho de eficácia quando utilizada a técnica de correlação. Isso se deve a elevada taxa da média de alarmes falsos, onde a adição da técnica reduziu significativamente essa taxa. Isso resultou em grande variação na média da precisão entre as duas versões do sistema. É importante notar também que devido a essa ocorrência, houve uma diferença maior na média da métrica de precisão devido ao uso de correlação, se comparado aos resultados do ataque anterior.

7.1.1.3 DoS *Smurf*

O ataque de DoS *smurf* apresentou melhores resultados para as quantidades de detectores de 20, 40, 80 e 100. No entanto a técnica de correlação melhorou a eficácia da detecção apenas para a quantidade de 20 detectores.

A seguir, as tabelas em conjunto com os gráficos em barra apresentam os resultados obtidos:

Tabela 32 – MAIS-IDS - DoS *Smurf*

20 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	97.544%	1.152%	3.823%	96.177%
STD	2.386%	5.023%	0.381%	0.381%
Max	98.114%	23.047%	3.923%	97.834%
Min	87.143%	0.0%	2.166%	96.077%

Tabela 33 – Abordagem Proposta - DoS *Smurf*

20 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	97.764%	0.723%	3.823%	96.177%
STD	1.427%	3.15%	0.381%	0.381%
Max	98.114%	14.453%	3.923%	97.834%
Min	91.543%	0.0%	2.166%	96.077%

Tabela 34 – MAIS-IDS - DoS *Smurf*

40 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	96.471%	0.0%	7.231%	92.769%
STD	4.881%	0.0%	10.002%	10.002%
Max	98.114%	0.0%	48.185%	96.136%
Min	76.486%	0.0%	3.864%	51.185%

Tabela 35 – Abordagem Proposta - DoS *Smurf*

40 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	96.471%	0.0%	7.231%	92.769%
STD	4.881%	0.0%	10.002%	10.002%
Max	98.114%	0.0%	48.185%	96.136%
Min	76.486%	0.0%	3.864%	51.185%

Tabela 36 – MAIS-IDS - DoS *Smurf*

80 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	97.781%	0.0%	4.546%	95.454%
STD	1.426%	0.0%	2.923%	2.923%
Max	98.686%	0.0%	12.237%	97.307%
Min	94.029%	0.0%	2.693%	87.763%

Tabela 37 – Abordagem Proposta - DoS *Smurf*

80 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	97.781%	0.0%	4.546%	95.454%
STD	1.426%	0.0%	2.923%	2.923%
Max	98.686%	0.0%	12.237%	97.307%
Min	94.029%	0.0%	2.693%	87.763%

Tabela 38 – MAIS-IDS - DoS *Smurf*

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	96.859%	0.0%	6.437%	93.563%
STD	3.099%	0.0%	6.351%	6.351%
Max	98.686%	0.0%	25.176%	97.307%
Min	87.714%	0.0%	2.693%	74.824%

Tabela 39 – Abordagem Proposta - DoS *Smurf*

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	96.859%	0.0%	6.437%	93.563%
STD	3.099%	0.0%	6.351%	6.351%
Max	98.686%	0.0%	25.176%	97.307%
Min	87.714%	0.0%	2.693%	74.824%

Para 20 detectores, a adição da técnica de correlação melhorou a média das métricas de precisão e falso positivo, onde o primeiro sofreu uma leve melhora e o último obteve uma redução de 37.2%. Já o desvio padrão não apresentou redução apenas para as métricas de falso negativo e taxa de detecção, pois elas não sofreram alterações. A redução no desvio padrão mostra que a técnica de correlação tornou os valores de precisão e falso positivo mais estáveis. Isso mostra que no decorrer de 20 execuções não houve grande desvio em relação a média para essas métricas.

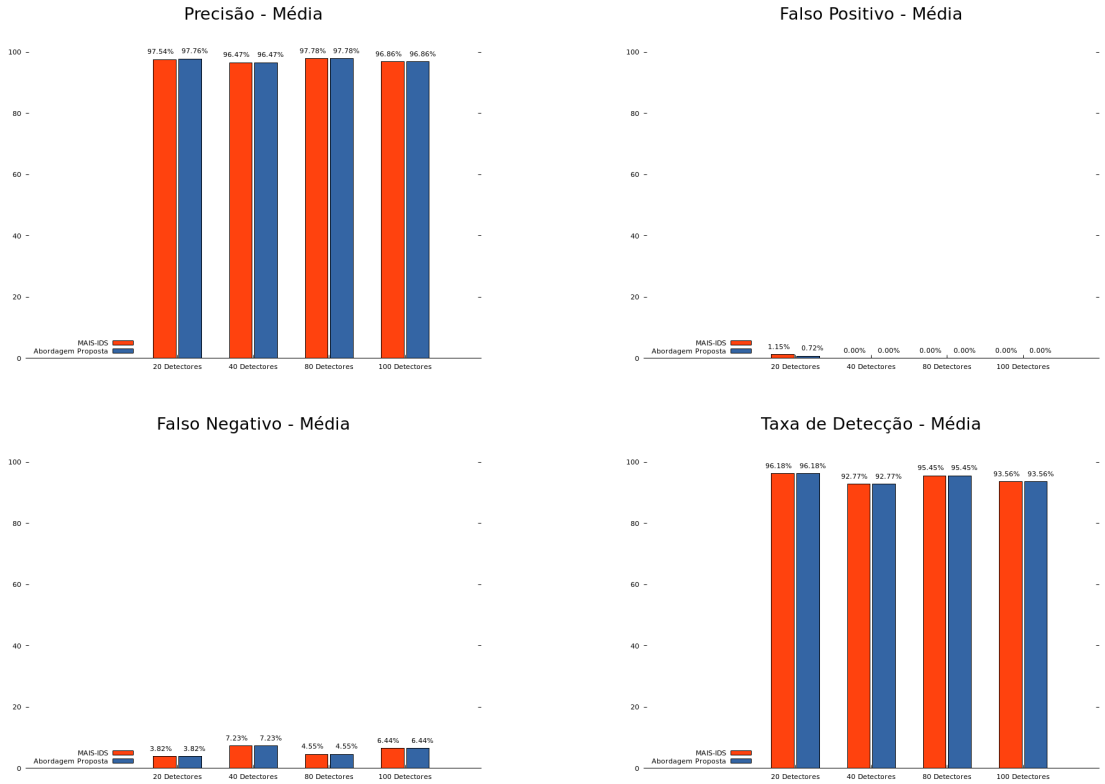


Figura 36 – DoS *Smurf* - Média

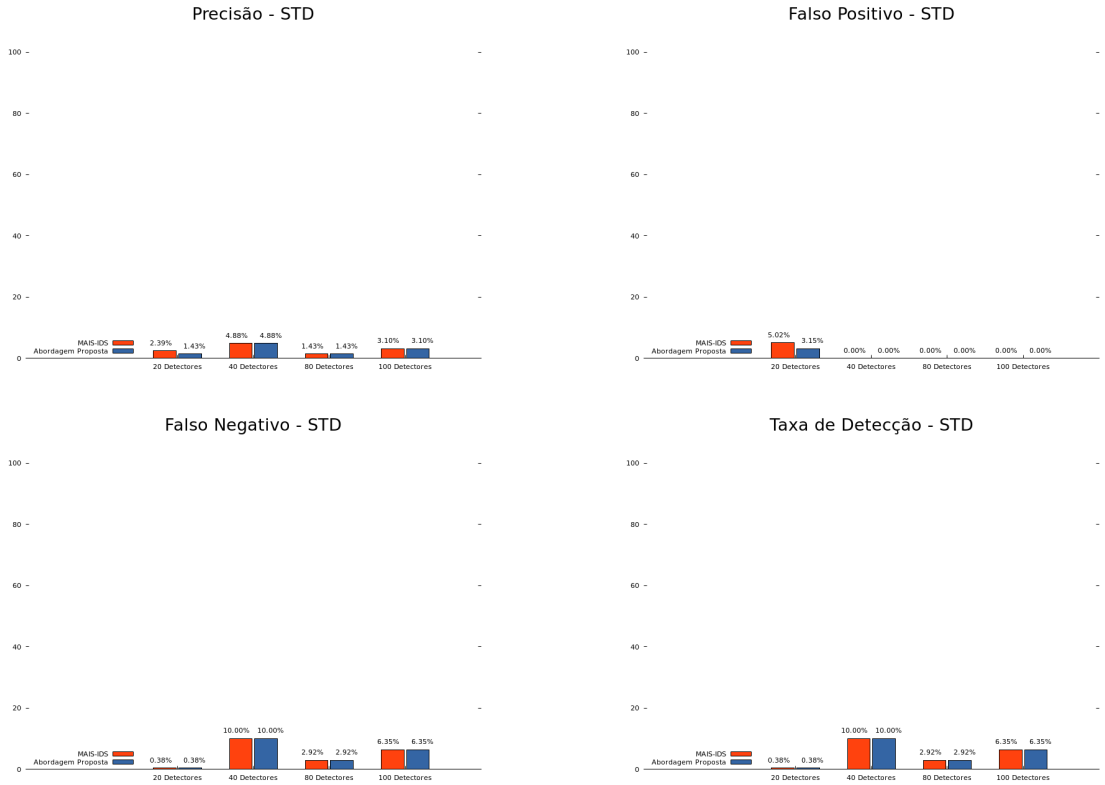


Figura 37 – DoS *Smurf* - STD

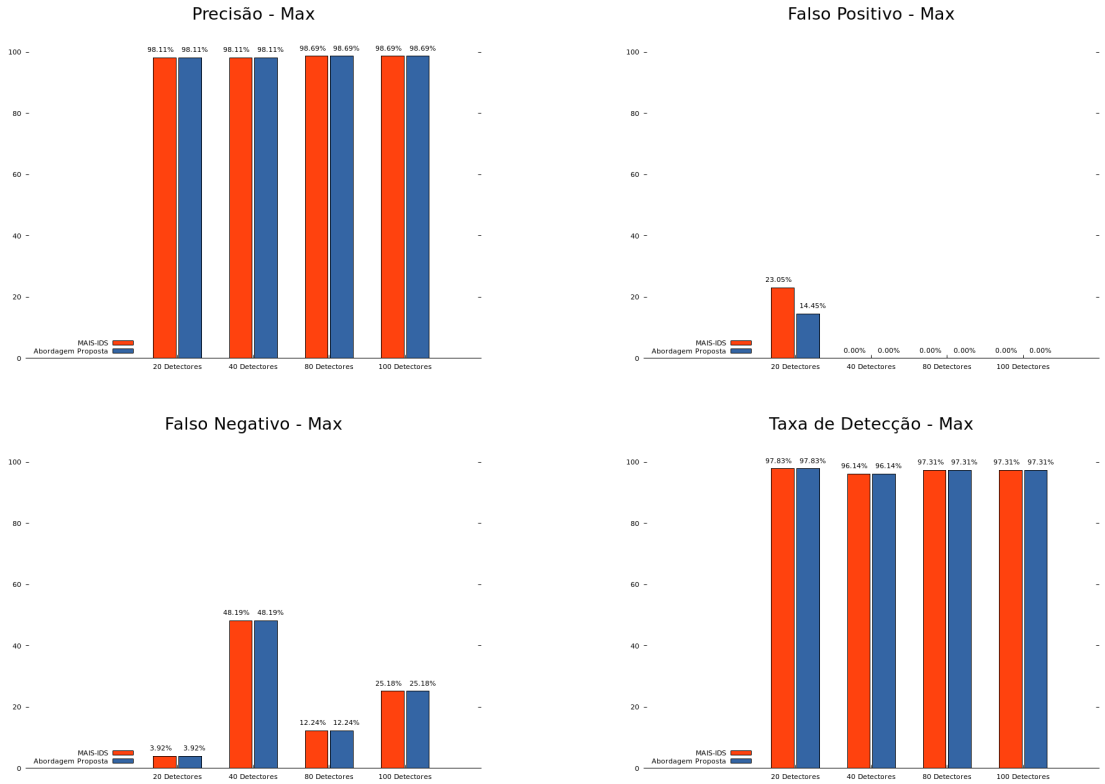


Figura 38 – DoS *Smurf* - Max

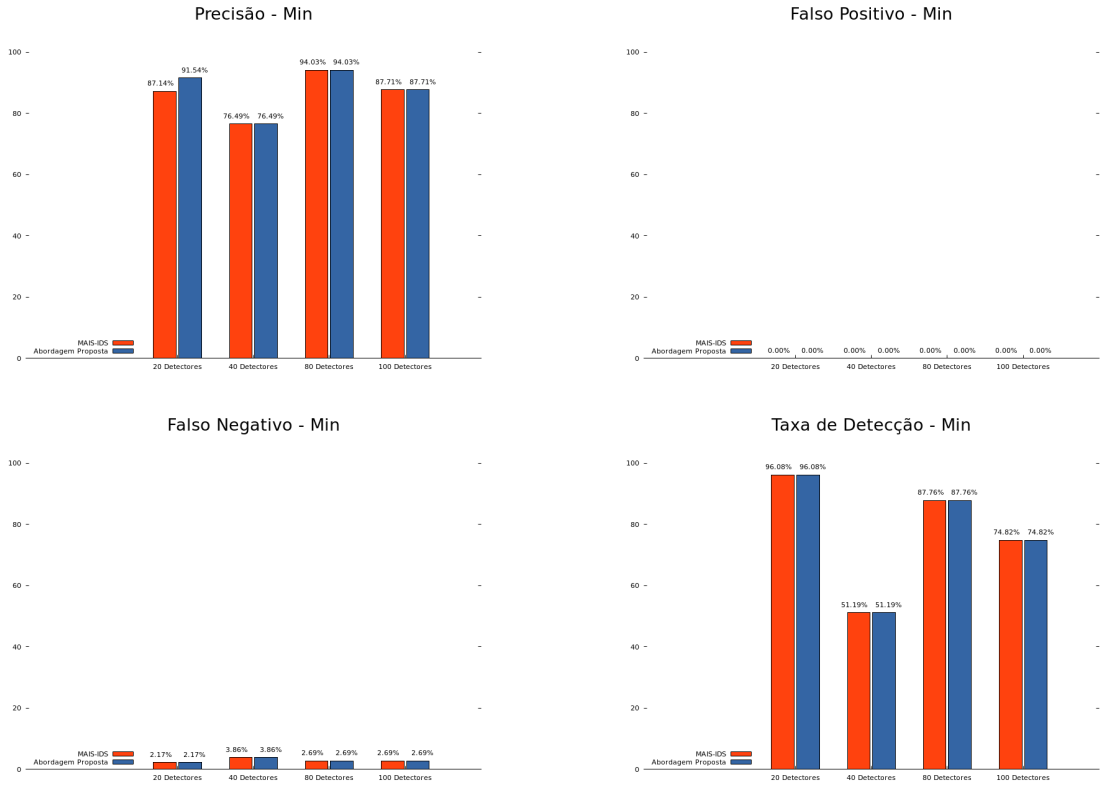


Figura 39 – DoS *Smurf* - Min

Quando testado com 40 detectores, nenhuma métrica sofreu alterações. Uma vez que o valor de alarmes falsos permaneceu sempre em 0.0%, a adição da técnica de correlação não melhorou a eficácia da detecção. No entanto, em relação ao cenário anterior, de 20 detectores, não houve redução apenas para a média da métrica de falso negativo, considerando ambas as abordagens, na presença e ausência de correlação. Já o desvio padrão apenas não sofreu acréscimo para a métrica de falso positivo, pois ela sempre permanece em 0.0% a partir de 40 detectores. Para valores máximos, o falso negativo sofre um aumento considerável, enquanto a taxa de detecção reduz, e a precisão continua a mesma. Referente a valores mínimos, a precisão e a taxa de detecção sofrem uma considerável redução, já a taxa de falso positivo permanece a mesma.

Quando o sistema é testado com 80 detectores, a adição de correlação de alertas não altera a eficácia. No entanto em relação ao cenário anterior, de 40 detectores, o sistema sofre uma leve melhora pois sua média de precisão aumenta. Já para 100 detectores, o cenário é semelhante aos dois anteriores, pois a adição da técnica de correlação não altera os resultados da detecção. Porém ao contrário desses cenários, o sistema começa a apresentar uma queda na eficácia pois sua precisão começa a diminuir.

De um modo geral, esse cenário de ataque apresentou resultados semelhantes ao de DoS Land, onde a média de alertas falsos foi de 0.0%. A exceção foi para a utilização de 20 detectores, onde ocorre uma leve queda no valor de falso positivo, e um leve aumento na precisão, devido ao método de correlação.

7.1.1.4 DoS *TCP Flood*

As quantidades de detectores para o qual esse tipo de ataque apresentou melhor eficácia foram 100, 200, 300 e 400. Para os resultados apresentados, a técnica de correlação surtiu efeito apenas a partir de 400 detectores, para números menores a adição de correlação não melhorou a eficácia da detecção, uma vez que não alterou os valores da métrica de falso positivo. Consequentemente, a adição da técnica não surtiu efeito algum na eficácia da detecção para essas quantidades.

A seguir, são apresentadas tabelas e figuras que exibem os resultados obtidos para cada quantidade de detectores para o qual o sistema foi testado:

Tabela 40 – MAIS-IDS - DoS *TCP Flood*

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	87.895%	3.634%	28.931%	71.069%
STD	4.242%	0.481%	11.864%	11.864%
Max	94.14%	3.965%	38.625%	88.0%
Min	84.429%	2.769%	12.0%	61.375%

Tabela 41 – Abordagem Proposta - DoS *TCP Flood*

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	87.895%	3.634%	28.931%	71.069%
STD	4.242%	0.481%	11.864%	11.864%
Max	94.14%	3.965%	38.625%	88.0%
Min	84.429%	2.769%	12.0%	61.375%

O sistema de detecção apresentou uma queda na eficácia de 100 para 200 detectores, pois houve uma redução na média da precisão. A razão para isso se deve ao aumento na média

Tabela 42 – MAIS-IDS - DoS *TCP Flood*

200 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	86.946%	3.93%	31.175%	68.825%
STD	4.174%	0.086%	12.356%	12.356%
Max	93.554%	3.965%	43.5%	88.25%
Min	82.796%	3.65%	11.75%	56.5%

Tabela 43 – Abordagem Proposta - DoS *TCP Flood*

200 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	86.946%	3.93%	31.175%	68.825%
STD	4.174%	0.086%	12.356%	12.356%
Max	93.554%	3.965%	43.5%	88.25%
Min	82.796%	3.65%	11.75%	56.5%

Tabela 44 – MAIS-IDS - DoS *TCP Flood*

300 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	92.438%	3.965%	14.706%	85.294%
STD	0.805%	0.0%	2.403%	2.403%
Max	93.721%	3.965%	16.625%	89.125%
Min	91.796%	3.965%	10.875%	83.375%

Tabela 45 – Abordagem Proposta - DoS *TCP Flood*

300 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	92.438%	3.965%	14.706%	85.294%
STD	0.805%	0.0%	2.403%	2.403%
Max	93.721%	3.965%	16.625%	89.125%
Min	91.796%	3.965%	10.875%	83.375%

Tabela 46 – MAIS-IDS - DoS *TCP Flood*

400 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	85.098%	17.033%	10.669%	89.331%
STD	9.859%	17.196%	5.441%	5.441%
Max	92.549%	58.024%	16.125%	99.0%
Min	61.072%	5.286%	1.0%	83.875%

Tabela 47 – Abordagem Proposta - DoS *TCP Flood*

400 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	85.866%	15.878%	10.669%	89.331%
STD	8.434%	15.068%	5.441%	5.441%
Max	92.549%	51.353%	16.125%	99.0%
Min	65.509%	5.286%	1.0%	83.875%

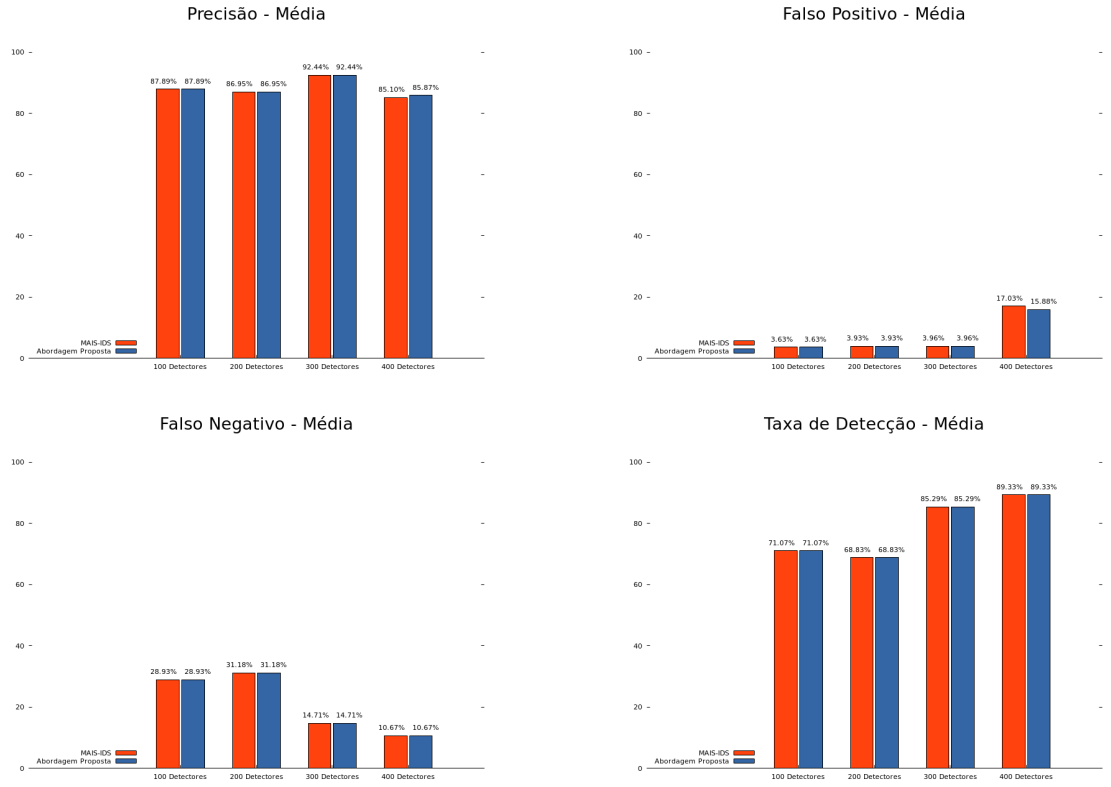


Figura 40 – DoS *TCP Flood* - Média

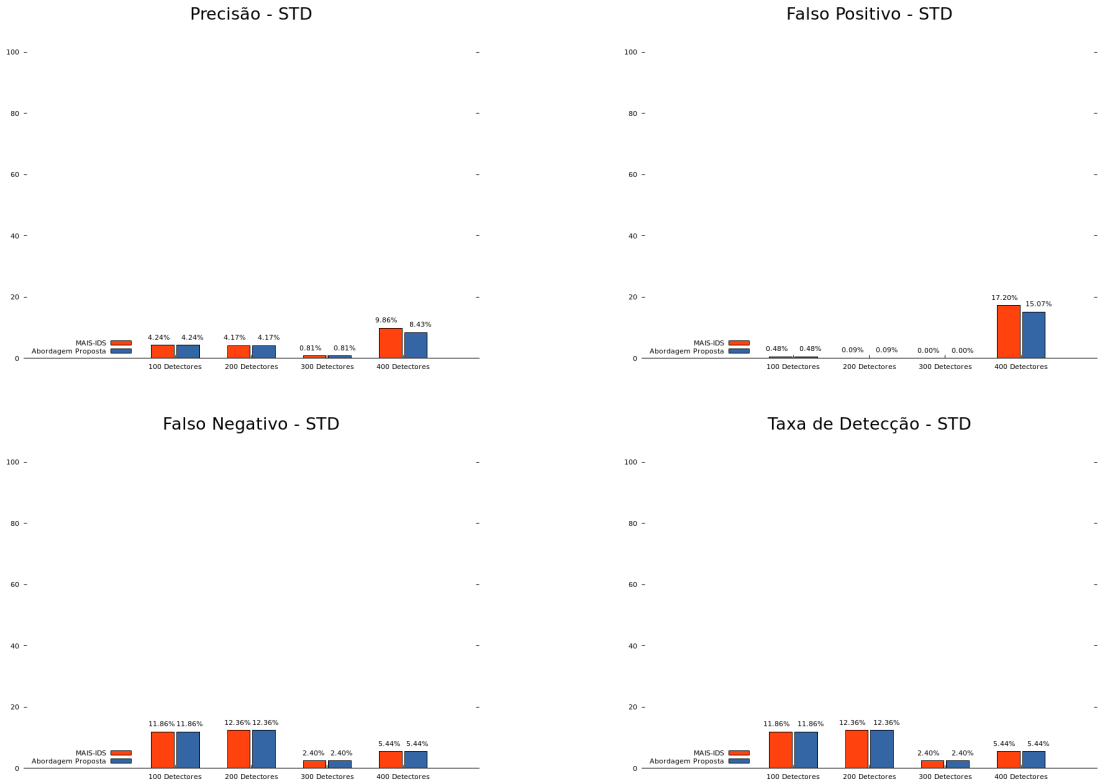


Figura 41 – DoS *TCP Flood* - STD

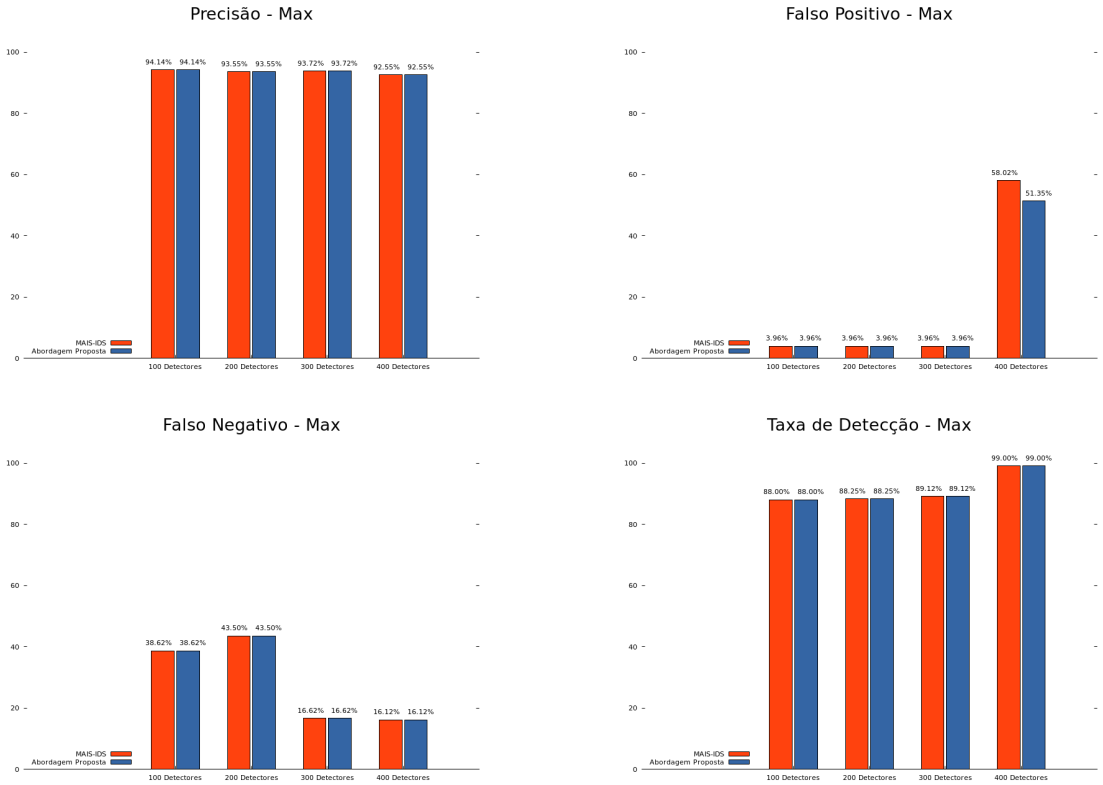


Figura 42 – DoS *TCP Flood* - Max

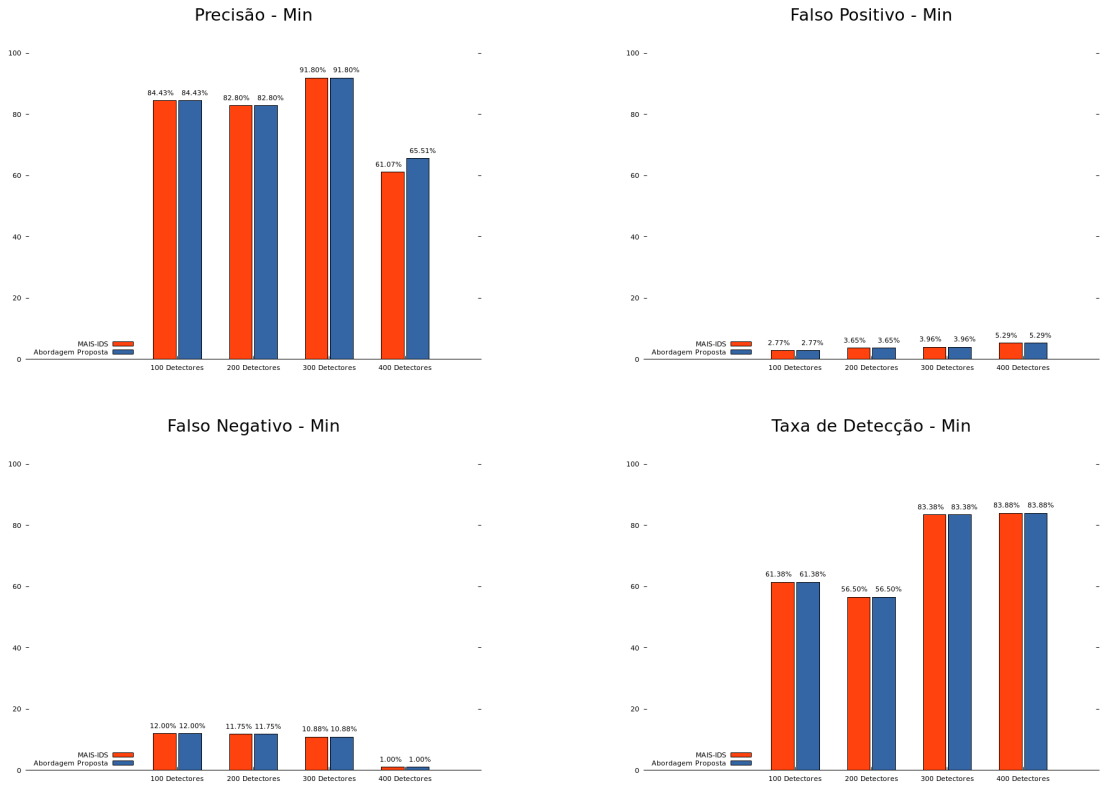


Figura 43 – DoS *TCP Flood* - Min

dos valores de falso positivo e negativo, e na redução da média da taxa de detecção. No entanto para as métricas de precisão e falso positivo, o desvio padrão foi menor, o que significa menor desvio em relação a média para essas medidas. Referente aos valores de falso negativo e taxa de detecção, houve um aumento no desvio padrão, o que demonstra maior instabilidade devido ao maior desvio da média obtida para essas métricas.

Enquanto para 300 detectores o sistema apresenta sua maior eficácia, pois se trata do cenário em que ele atinge sua maior média de precisão, alcançando o valor de 92.438%. Porém, para a quantidade de 400 detectores, ocorre uma redução na eficácia da detecção. No entanto, para esse cenário a técnica de correlação começa a surtir efeito e eleva a média da precisão de 85.098% para 85.866%. Isso se deve a uma redução de 6.8% na taxa de falso positivo.

Para esse cenário de ataque, apesar do sistema apresentar uma melhora na eficácia da detecção devido a técnica de correlação a partir de 400 detectores, ele obteve sua melhor eficácia para a quantidade de 300. É possível que para esse cenário específico, o aumento dessa quantidade resultou no acréscimo da taxa de alertas falsos, que pôde ser reduzida pela adição do método de correlação. Isso resultou em uma melhora na eficácia do sistema, uma vez que a redução na taxa de falso positivo foi de apenas 6.8%.

7.1.1.5 DoS *Teardrop*

Nesse cenário de ataque o sistema apresenta sua melhor eficácia para as quantidades de detectores de 20, 40, 80 e 100. Para esse ataque, a técnica de correlação melhorou a eficácia da detecção para todas as quantidades de detectores.

As tabelas abaixo exibem os resultados obtidos em conjunto com os gráficos de barra:

Tabela 48 – MAIS-IDS - DoS *Teardrop*

20 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	88.248%	1.08%	32.355%	67.645%
STD	2.615%	2.273%	5.786%	5.786%
Max	91.041%	7.8%	45.56%	73.745%
Min	81.818%	0.0%	26.255%	54.44%

Tabela 49 – Abordagem Proposta - DoS *Teardrop*

20 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	88.847%	0.07%	32.548%	67.452%
STD	2.011%	0.23%	5.895%	5.895%
Max	91.041%	1.0%	45.946%	73.745%
Min	84.321%	0.0%	26.255%	54.054%

Tabela 50 – MAIS-IDS - DoS *Teardrop*

40 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	90.026%	1.32%	26.68%	73.32%
STD	1.708%	2.133%	3.132%	3.132%
Max	92.095%	6.6%	33.591%	77.606%
Min	86.298%	0.0%	22.394%	66.409%

Tabela 51 – Abordagem Proposta - DoS *Teardrop*

40 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	90.58%	0.43%	26.776%	73.224%
STD	1.274%	0.976%	3.185%	3.185%
Max	92.227%	3.8%	33.591%	77.22%
Min	87.879%	0.0%	22.78%	66.409%

Tabela 52 – MAIS-IDS - DoS *Teardrop*

80 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	91.574%	0.15%	24.402%	75.598%
STD	0.508%	0.424%	1.13%	1.13%
Max	92.227%	1.6%	26.255%	77.22%
Min	90.382%	0.0%	22.78%	73.745%

Tabela 53 – Abordagem Proposta - DoS *Teardrop*

80 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	91.627%	0.0%	24.537%	75.463%
STD	0.442%	0.0%	1.295%	1.295%
Max	92.227%	0.0%	27.799%	77.22%
Min	90.514%	0.0%	22.78%	72.201%

Tabela 54 – MAIS-IDS - DoS *Teardrop*

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	90.685%	1.57%	24.266%	75.734%
STD	2.209%	3.24%	1.247%	1.247%
Max	92.227%	11.8%	25.483%	78.378%
Min	83.663%	0.0%	21.622%	74.517%

Tabela 55 – Abordagem Proposta - DoS *Teardrop*

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	91.403%	0.32%	24.575%	75.425%
STD	0.821%	0.987%	1.147%	1.147%
Max	92.227%	4.4%	25.483%	77.992%
Min	88.406%	0.0%	22.008%	74.517%

Para a quantidade de 20 detectores, referente as médias obtidas, o sistema apresentou uma leve melhora na precisão. Esse fato ocorreu devido a uma redução de 93.5% na taxa de alarmes falsos. Contudo, ocorreu uma pequena queda na taxa de detecção e aumento no valor de falso negativo. Já em relação ao desvio padrão, a adição do método de correlação aumentou a estabilidade das métricas de precisão e falso positivo, pois essas métricas se desviaram menos da média. Em seguida, houve uma maior instabilidade para valores de falso negativo e taxa de detecção. Para valores máximos, o modelo proposto apresentou melhorias na métrica de falso positivo, seguido de uma piora em relação ao valor de falso negativo, enquanto a precisão não se alterou. Enquanto para valores mínimos, a abordagem proposta aumentou a precisão, e reduziu o

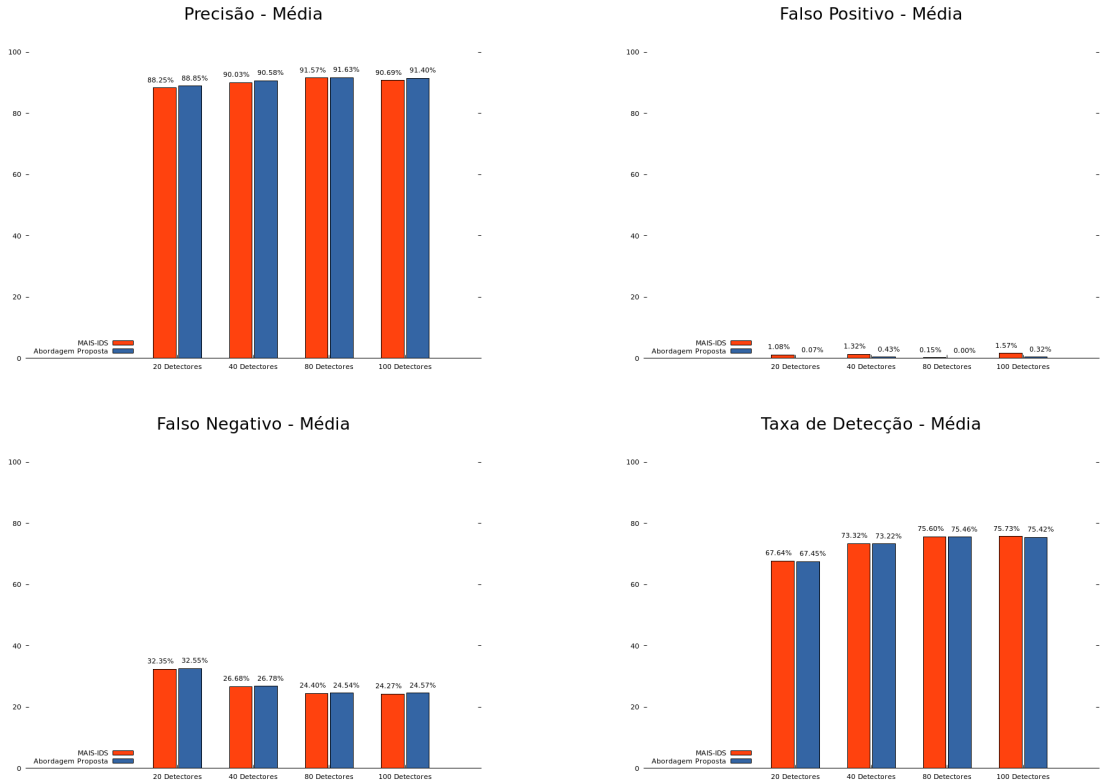


Figura 44 – DoS *Teardrop* - Média

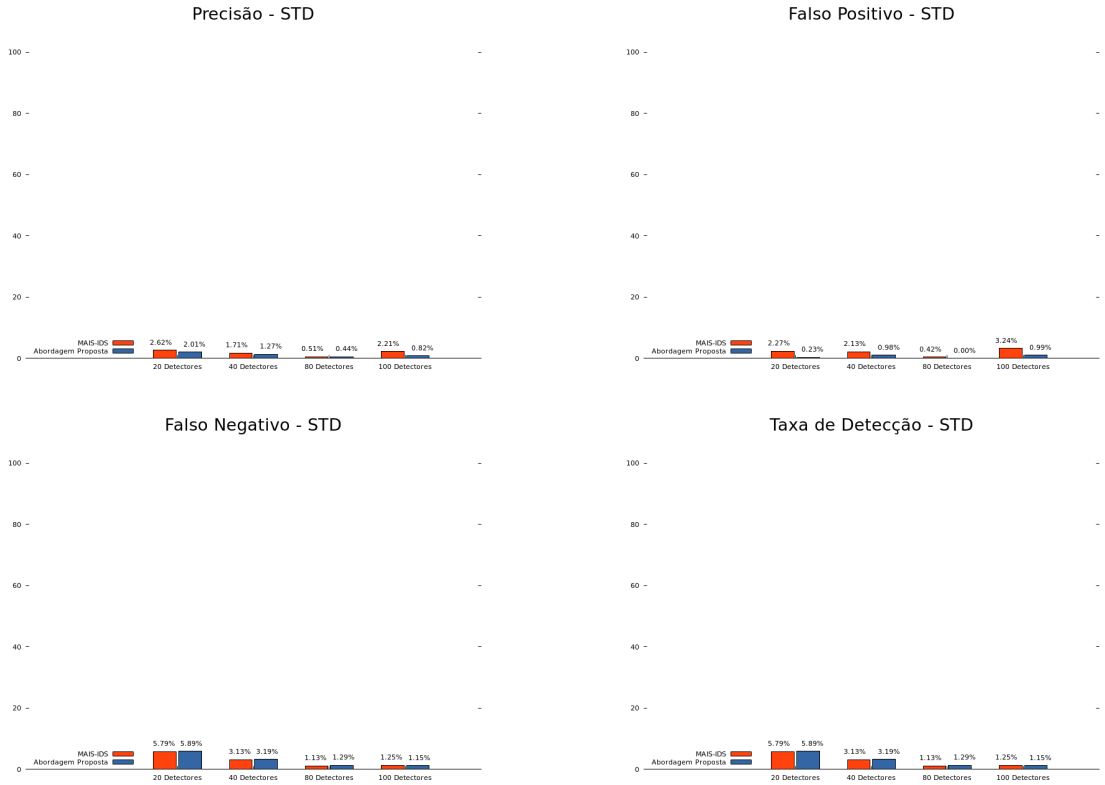


Figura 45 – DoS *Teardrop* - STD

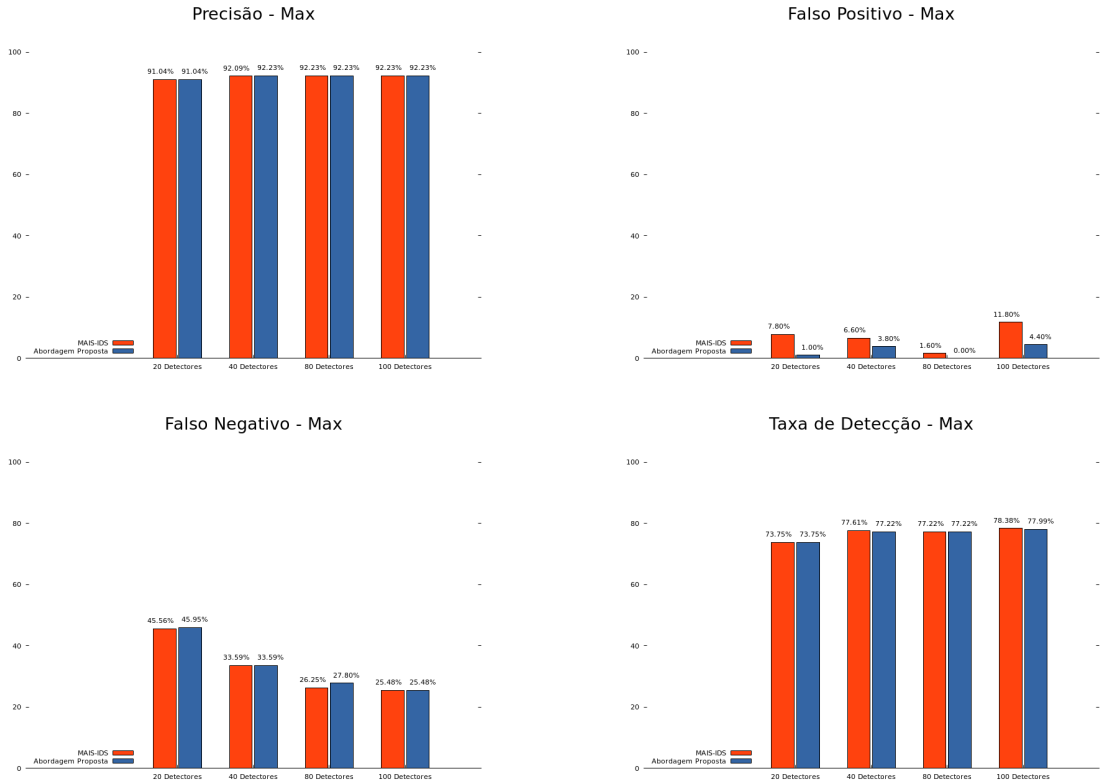


Figura 46 – DoS *Teardrop* - Max

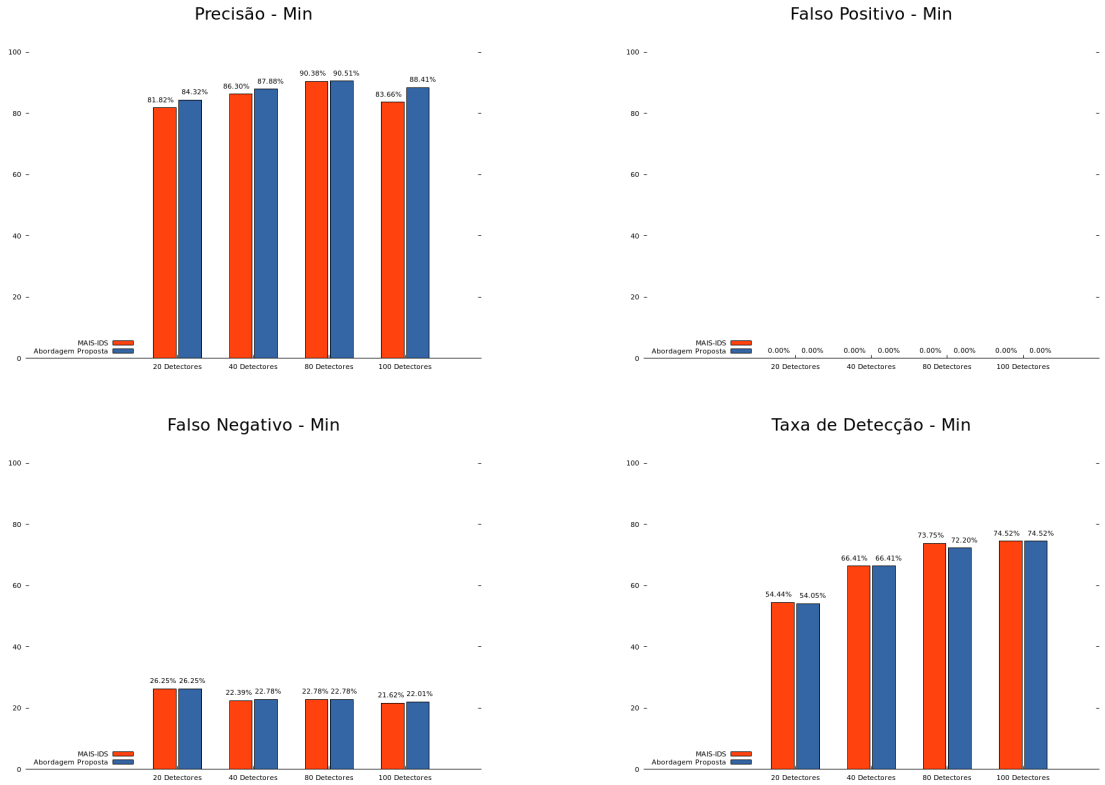


Figura 47 – DoS *Teardrop* - Min

valor da taxa de detecção, enquanto as métricas de falso positivo e falso negativo não se alteraram. Quando testado com 40 detectores, houve um comportamento semelhante referente a média das métricas. O sistema melhorou levemente a sua precisão, devido a uma redução de 67.4% nos alertas falsos, o que não acrescentou muito a precisão pois a média de alertas falsos apresentou um valor baixo. Assim como para a quantidade de 20 detectores, o sistema sofreu uma leve queda na taxa de detecção. Em relação ao desvio padrão a abordagem proposta apresentou maior estabilidade para as métricas de precisão e falso positivo, enquanto reduziu a estabilidade para falso negativo e taxa de detecção. Em relação aos valores máximos, a precisão aumentou enquanto a quantidade de falso positivo diminuiu e a de falso negativo permaneceu constante, já para a taxa de detecção houve uma pequena queda. Para valores mínimos a abordagem proposta apresentou melhores resultados apenas para a precisão.

Para 80 detectores houve melhoras nas médias da precisão, e falso positivo. Além disso, é importante notar que a taxa de alertas falsos permanece sempre em 0.0%. Isso significa que no decorrer de 20 execuções, a redução na taxa de falso positivo foi sempre de 100%. Já como na maioria dos casos, o modelo proposto aumentou a estabilidade para a precisão e valores de falso positivo. Para a métrica de falso negativo e taxa de detecção, houve um crescimento na instabilidade referente ao desvio padrão. Para valores mínimos, a adição de correlação melhorou a precisão, enquanto houve uma pequena redução na taxa de detecção, e as outras métricas permaneceram estáveis.

Para 100 detectores os resultados foram semelhantes, onde a redução na média da taxa de falso positivo foi de 79.6%. Já em relação ao desvio padrão o modelo proposto apresentou maior estabilidade para todas as métricas. Enquanto para valores máximos, o sistema apresentou melhor resultado para o valor de falso positivo, e pior valor para a taxa de detecção, enquanto as métricas de precisão e falso negativo permaneceram constantes. Para valores mínimos houve uma melhora na precisão e um aumento na taxa de falso negativo, enquanto as taxas de falso positivo e de detecção permaneceram as mesmas.

Para esse cenário é possível perceber que a adição da técnica de correlação resultou em uma melhor eficácia para todas as quantidades de detectores para o qual o sistema foi testado. Houve um pequeno aumento na precisão para todos os casos, sem nenhum custo adicional. A razão para isso se trata de uma média baixa de alertas falsos quando o sistema é utilizado sem o auxílio da técnica de correlação. Isso resulta em um baixo aumento na eficácia ao se adicionar esse método.

7.1.1.6 DDoS HTTP

Em relação ao ataque de negação de serviço distribuído por meio do protocolo HTTP, o sistema de detecção apresentou os melhores resultados para as quantidades de 100, 200, 300 e 400 detectores. Para quantidades a partir de 200, a adição de correlação melhorou a eficácia do sistema. Além disso, as taxas de detecção e falso negativo não foram afetadas pela adição da

técnica em nenhum cenário.

A seguir são apresentadas tabelas e gráficos de barra que descrevem a eficácia do sistema para todas as quantidades de detectores para o qual ele foi testado.

Tabela 56 – MAIS-IDS - DDoS HTTP

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	82.401%	0.935%	61.733%	38.267%
STD	3.769%	1.063%	15.97%	15.97%
Max	90.726%	4.154%	71.333%	77.0%
Min	80.037%	0.566%	23.0%	28.667%

Tabela 57 – Abordagem Proposta - DDoS HTTP

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	82.401%	0.935%	61.733%	38.267%
STD	3.769%	1.063%	15.97%	15.97%
Max	90.726%	4.154%	71.333%	77.0%
Min	80.037%	0.566%	23.0%	28.667%

Tabela 58 – MAIS-IDS - DDoS HTTP

200 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	86.508%	7.627%	29.025%	70.975%
STD	1.618%	2.954%	6.249%	6.249%
Max	90.589%	9.377%	47.5%	76.333%
Min	85.656%	0.692%	23.667%	52.5%

Tabela 59 – Abordagem Proposta - DDoS HTTP

200 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	87.229%	6.633%	29.025%	70.975%
STD	1.298%	2.404%	6.249%	6.249%
Max	90.589%	7.992%	47.5%	76.333%
Min	86.478%	0.692%	23.667%	52.5%

Tabela 60 – MAIS-IDS - DDoS HTTP

300 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	86.446%	8.874%	25.95%	74.05%
STD	1.657%	2.342%	0.205%	0.205%
Max	89.813%	10.321%	26.333%	74.333%
Min	85.427%	4.216%	25.667%	73.667%

Tabela 61 – Abordagem Proposta - DDoS HTTP

300 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	87.25%	7.766%	25.95%	74.05%
STD	1.257%	1.793%	0.205%	0.205%
Max	89.813%	8.936%	26.333%	74.333%
Min	86.432%	4.216%	25.667%	73.667%

Tabela 62 – MAIS-IDS - DDoS HTTP

400 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	85.831%	9.78%	25.792%	74.208%
STD	0.294%	0.211%	0.652%	0.652%
Max	87.072%	10.132%	26.0%	76.667%
Min	85.656%	8.999%	23.333%	74.0%

Tabela 63 – Abordagem Proposta - DDoS HTTP

400 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	86.802%	8.442%	25.792%	74.208%
STD	0.153%	0.115%	0.652%	0.652%
Max	87.392%	8.748%	26.0%	76.667%
Min	86.661%	8.307%	23.333%	74.0%

A adição da técnica de correlação para os detectores finais gerados não alterou nenhuma métrica de detecção referente ao primeiro cenário de 100 detectores. No entanto, a partir de 200, a técnica reduziu a média de alertas falsos em 13.0% e aumentou a média da precisão de 86,508% para 87,229%. Em relação ao desvio padrão, as métricas de precisão e falso positivo reduziram seus valores. Além disso, para os valores máximos, a adição de correlação diminuiu o valor de falso positivo alcançado, enquanto a precisão se manteve inalterada. Já para valores mínimos, a precisão aumentou enquanto a quantidade de alertas falsos e a taxa de detecção se mantiveram inalteradas.

Para a quantidade de 300 detectores o sistema apresentou variações da eficácia semelhantes ao cenário anterior. A adição de correlação melhorou a média das métricas de precisão e falso positivo sem nenhum custo adicional, onde os alertas falsos foram reduzidos em 12.5%.

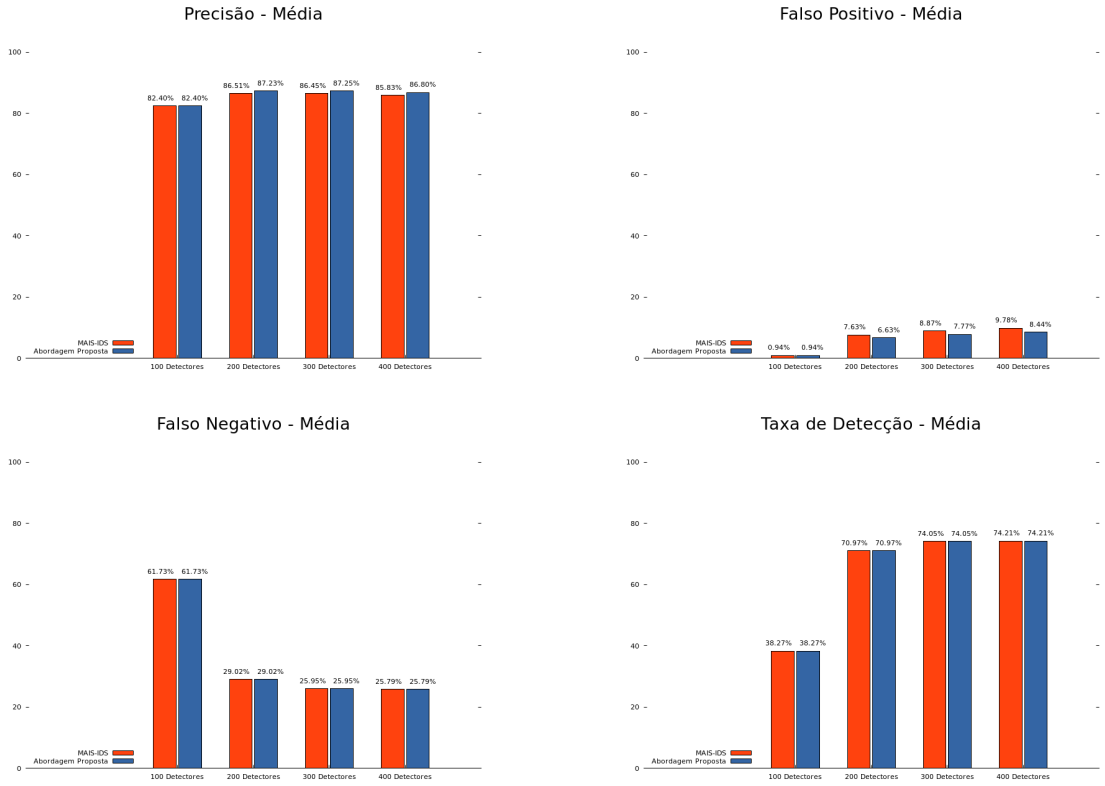


Figura 48 – DDoS HTTP - Média

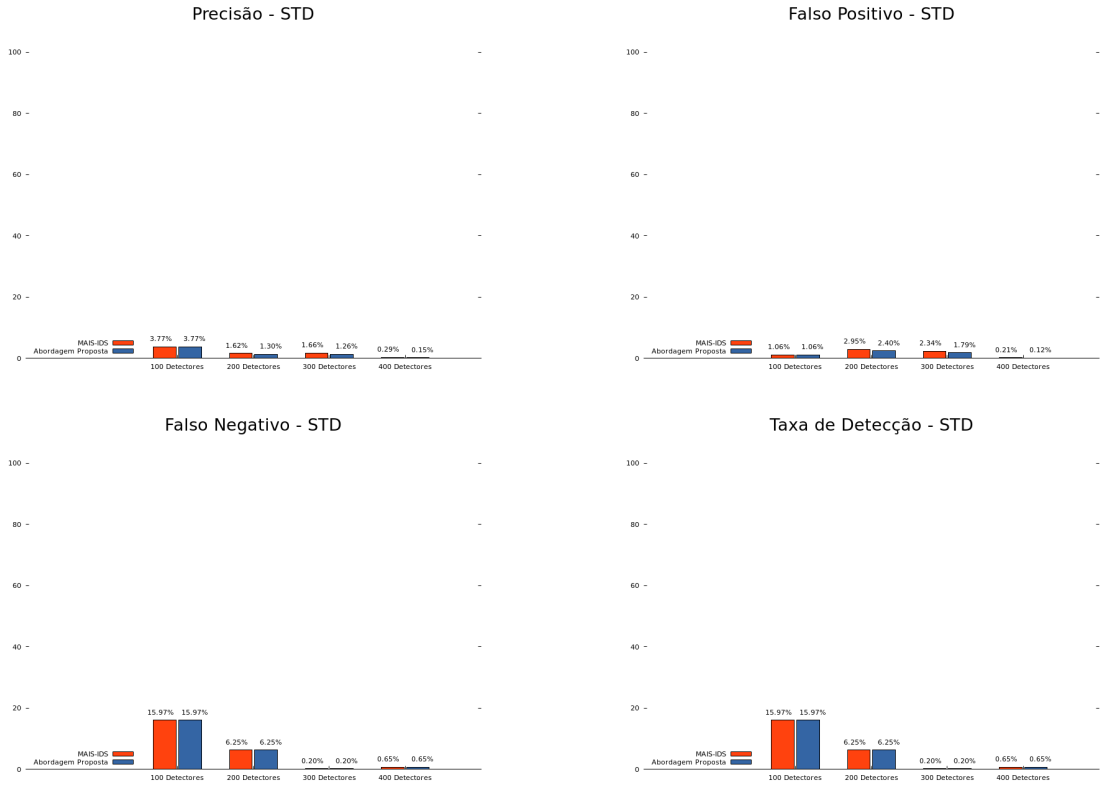


Figura 49 – DDoS HTTP - STD

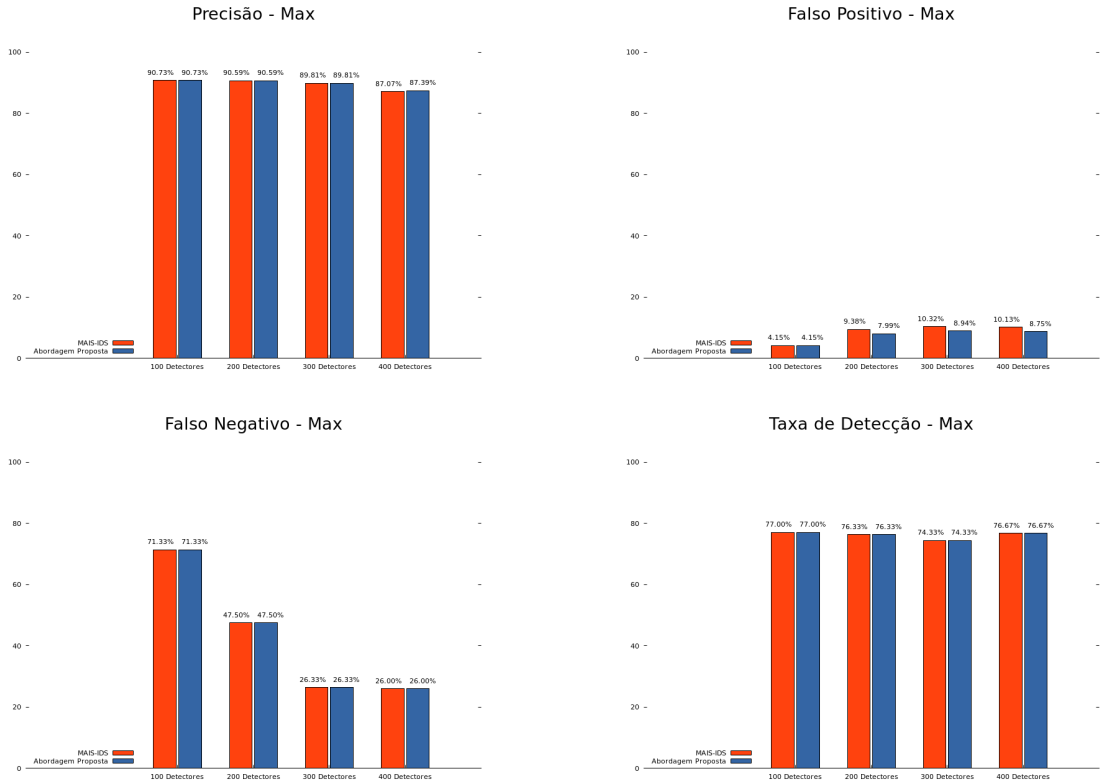


Figura 50 – DDoS HTTP - Max

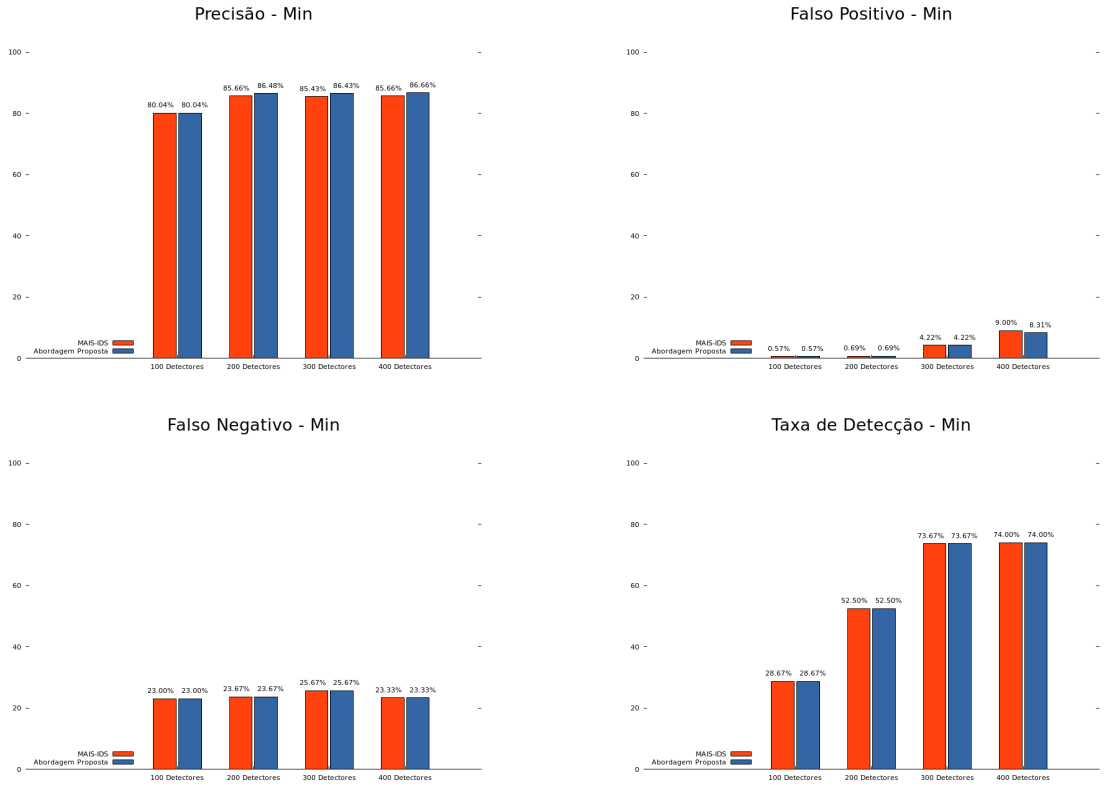


Figura 51 – DDoS HTTP - Min

Já para a quantidade de 400 detectores houve uma queda na eficácia do sistema referente aos resultados do cenário anterior, considerando as versões correspondentes do sistema na presença e ausência de correlação. Apesar disso, para 400 detectores a eficácia da detecção aumentou devido ao método de correlação, com resultados semelhantes ao dois anteriores.

Para esse tipo de ataque, o sistema apresentou resultados inferiores a 90% em sua média de precisão por se tratar de um ataque de negação de serviço focado para a camada de aplicação, pois ameaças desse tipo são mas difíceis de detectar (LIM et al., 2014). Isso se deve a uma menor alteração nos dados estatísticos do tráfego de rede, possibilitando que ameaças desse tipo se assemelhem a tráfego normal (LIM et al., 2014). Apesar disso, podemos perceber que de forma semelhante aos outros cenários, quando a técnica de correlação é adicionada, a precisão aumenta, seguida de uma redução na taxa de alarmes falsos. Para esse caso específico, o sistema não apresentou melhor eficácia apenas para 100 detectores. Também é curioso notar que conforme a quantidade de detectores aumentava o desvio padrão reduzia, tanto na presença quanto na ausência de correlação.

7.1.1.7 DDoS *Slowloris*

Ataques de negação de serviço conhecidos como *Slowloris* focam em serviços específicos na camada de aplicação, como *Apache*, por exemplo (Imperva, 2018). Para esse tipo de ataque, a eficácia da detecção exibiu melhores resultados para 100, 200, 300 e 400 detectores. A partir de 200 detectores a técnica de correlação melhora a eficácia da detecção. No entanto, para nenhum dos cenários testados as métricas de falso negativo e taxa de detecção sofrem alteração. As tabelas e gráficos de barra apresentam os resultados obtidos para esse tipo de ataque.

Tabela 64 – MAIS-IDS - DDoS *Slowloris*

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	95.704%	0.0%	8.973%	91.027%
STD	21.019%	0.0%	4.217%	4.217%
Max	97.731%	0.0%	20.778%	95.261%
Min	90.052%	0.0%	4.739%	79.222%

Tabela 65 – Abordagem Proposta - DDoS *Slowloris*

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	95.704%	0.0%	8.973%	91.027%
STD	21.019%	0.0%	4.217%	4.217%
Max	97.731%	0.0%	20.778%	95.261%
Min	90.052%	0.0%	4.739%	79.222%

Tabela 66 – MAIS-IDS - DDoS *Slowloris*

200 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	97.535%	0.162%	4.973%	95.027%
STD	0.473%	0.643%	1.251%	1.251%
Max	98.662%	2.958%	8.019%	98.299%
Min	96.161%	0.0%	1.701%	91.981%

Tabela 67 – Abordagem Proposta - DDoS *Slowloris*

200 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	97.536%	0.159%	4.973%	95.027%
STD	0.476%	0.643%	1.251%	1.251%
Max	98.691%	2.958%	8.019%	98.299%
Min	96.161%	0.0%	1.701%	91.981%

Para a quantidade de 100 detectores a adição de correlação não alterou a eficácia da detecção, não houve variação em nenhuma das métricas. Contudo, a partir de 200 detectores a técnica começa a melhorar a eficácia do sistema. Para essa quantidade houve um pequeno aumento na média da precisão, seguida de uma leve redução na média de falso positivo. Em relação

Tabela 68 – MAIS-IDS - DDoS *Slowloris*

300 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	97.161%	1.172%	4.654%	95.346%
STD	2.227%	3.821%	0.583%	0.583%
Max	98.168%	16.741%	6.865%	96.659%
Min	87.987%	0.0%	3.341%	93.135%

Tabela 69 – Abordagem Proposta - DDoS *Slowloris*

300 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	97.382%	0.748%	4.654%	95.346%
STD	1.365%	2.214%	0.583%	0.583%
Max	98.226%	8.371%	6.865%	96.659%
Min	92.35%	0.0%	3.341%	93.135%

Tabela 70 – MAIS-IDS - DDoS *Slowloris*

400 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	97.901%	0.22%	4.143%	95.857%
STD	0.44%	0.731%	1.233%	1.233%
Max	99.418%	3.348%	5.711%	99.089%
Min	97.15%	0.0%	0.911%	94.289%

Tabela 71 – Abordagem Proposta - DDoS *Slowloris*

400 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	97.913%	0.198%	4.143%	95.857%
STD	0.454%	0.729%	1.233%	1.233%
Max	99.506%	3.348%	5.711%	99.089%
Min	97.237%	0.0%	0.911%	94.289%

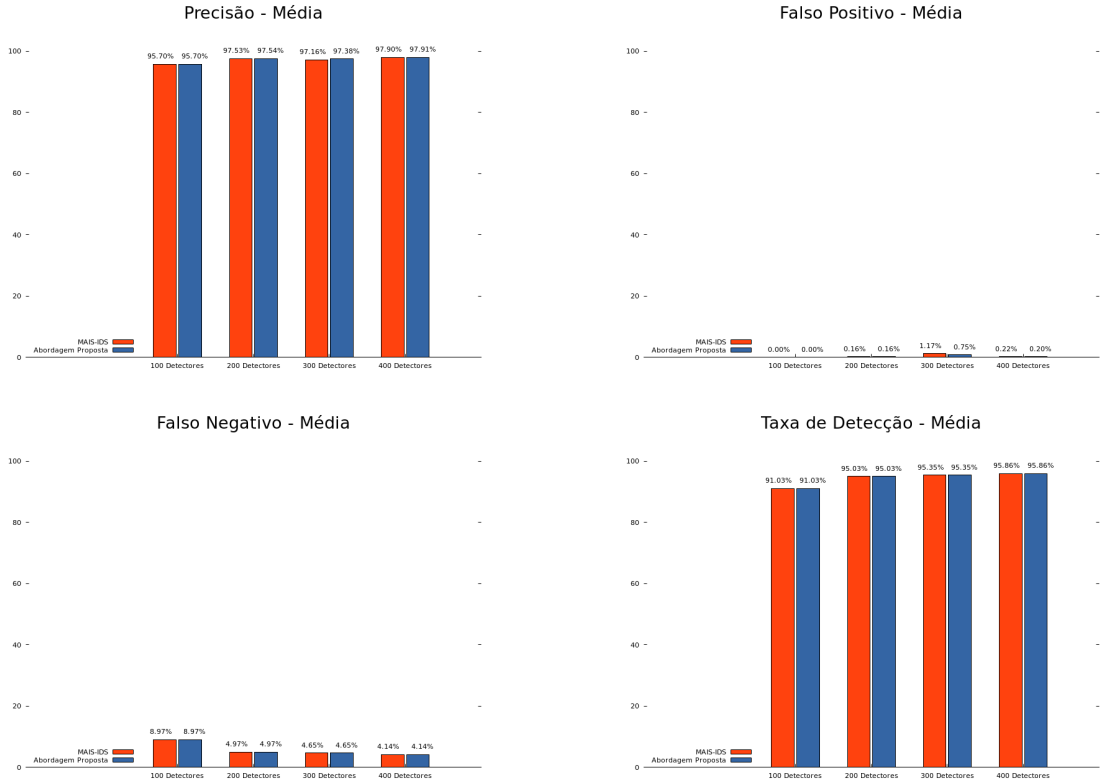


Figura 52 – DDoS *Slowloris* - Média

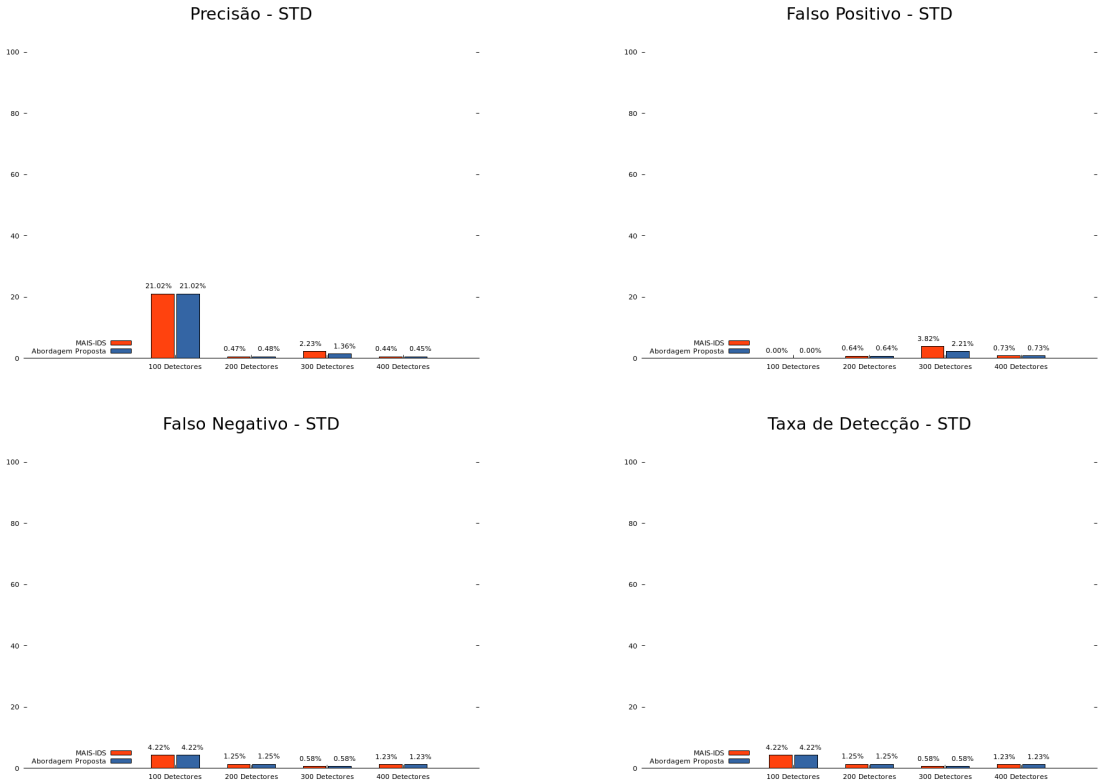


Figura 53 – DDoS *Slowloris* - STD

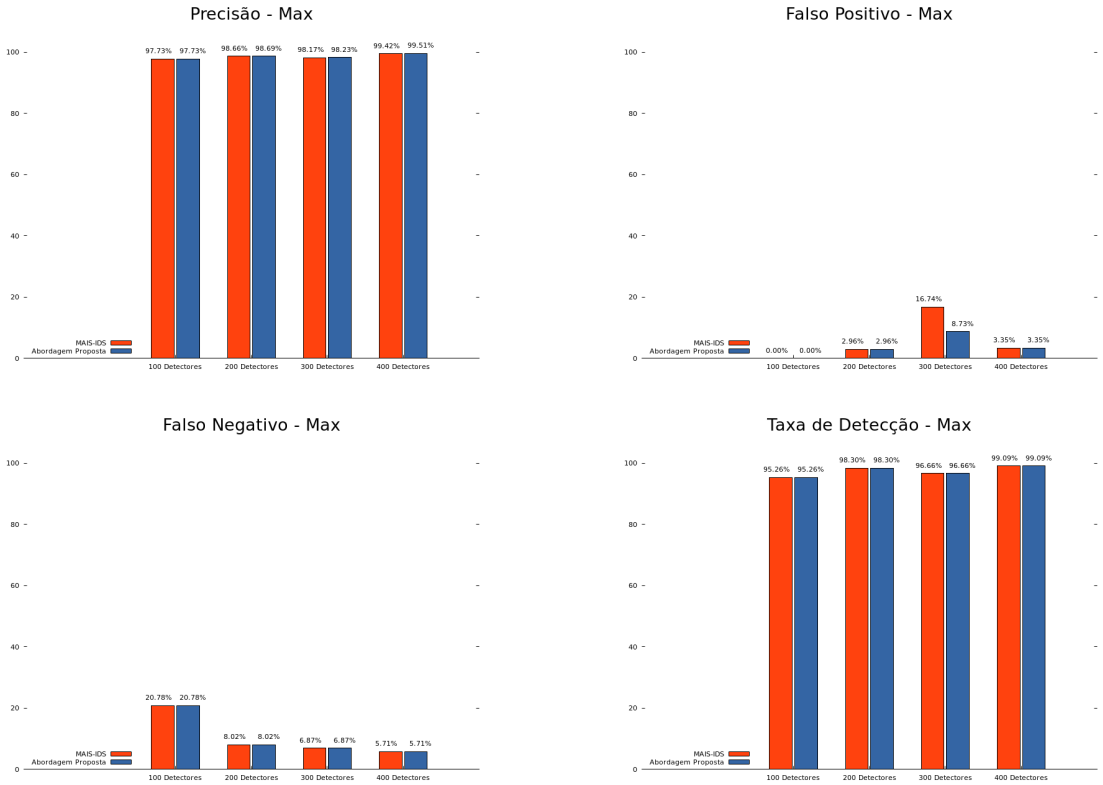


Figura 54 – DDoS *Slowloris* - Max

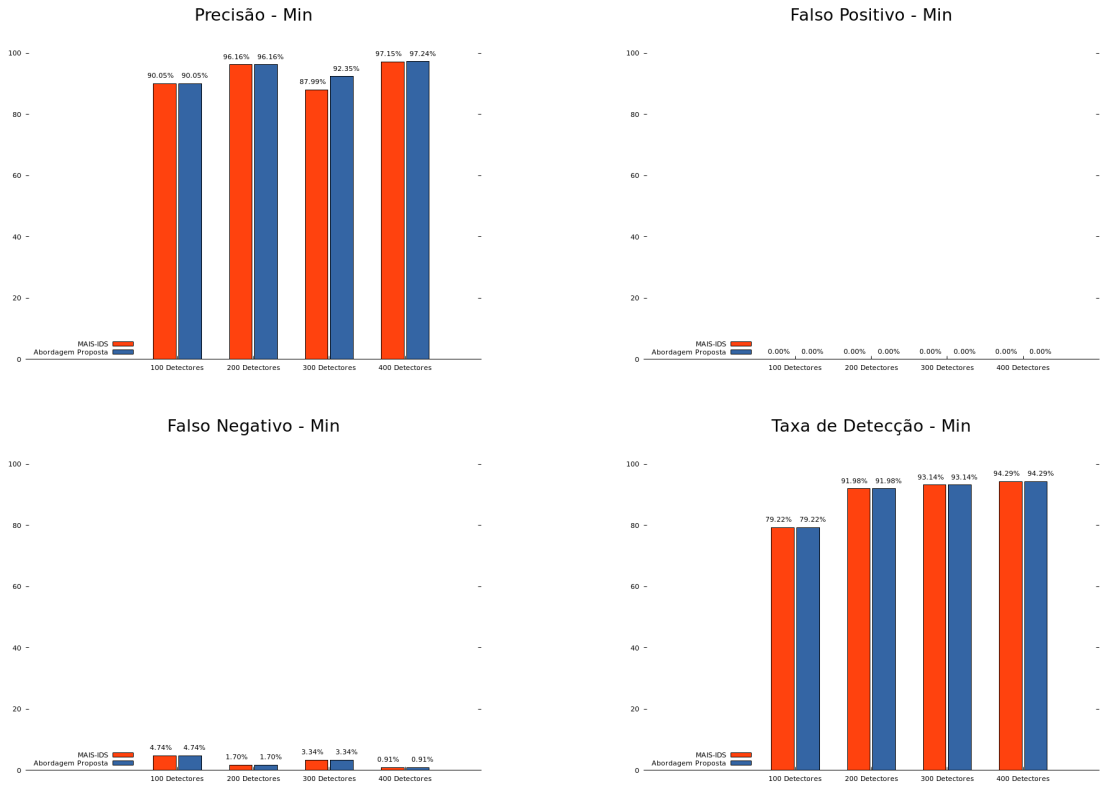


Figura 55 – DDoS *Slowloris* - Min

ao desvio padrão ocorreu um aumento na precisão, pois essa métrica alcançou um valor máximo de 98.691% contra a precisão de 98.662% referente ao sistema sem a técnica de correlação, onde o valor mínimo permaneceu constante. Já o desvio padrão de alertas falsos também permaneceu inalterado, uma vez que a variação entre valores máximo e mínimo permaneceu constante, contribuindo para essa ocorrência. Já em relação a taxa de detecção, não houve alterações nos seus respectivos valores máximo e mínimo.

A partir de 300 detectores o sistema continuou apresentando melhoria na eficácia da detecção. Para essa quantidade a média de precisão aumentou enquanto a de falso positivo reduziu. No entanto, em relação a precisão o valor máximo atingido foi maior enquanto o mínimo também sofreu acréscimo, e a redução na diferença entre os dois contribuiu na redução do desvio padrão. Isso demonstra que na maior parte das 20 execuções a técnica de correlação melhorou a precisão. Em relação a taxa de alertas falsos também houve uma redução na média e no desvio padrão, pois o valor máximo atingido diminuiu enquanto o mínimo permaneceu inalterado em 0.0%.

Para a quantidade de 400 detectores o sistema apresentou sua melhor eficácia tanto na utilização de correlação quanto na sua ausência. Referente a média da precisão, houve um leve aumento seguido de uma queda no desvio padrão. Além disso na variação entre os valores máximos e mínimos atingidos, a precisão alcançou 99.506% contra 99.418% sem correlação, enquanto seu valor mínimo atingido chegou a 97.237% contra 97.15%. Ocorreu também uma

pequena redução na média da taxa de alertas falsos, seguida de uma redução no seu desvio padrão, apesar da variação entre os valores mínimo e máximo permaneceram inalterados.

Apesar de se tratar de um ataque de negação de serviço focado a camada de aplicação, para esse cenário, o sistema apresentou uma eficácia média de precisão superior a 97%. A adição da técnica de correlação melhorou a eficácia para valores máximos de precisão de 99.506%, onde a média dessa métrica alcançou seu maior valor em 97.913%. É possível perceber também que em relação a quantidade de 100 detectores, todas as outras quantidades apresentaram um desvio padrão menor, com exceção da métrica de falso positivo, onde a média para 100 detectores foi de 0.0%.

7.1.2 Ataques de Reconhecimento - *Probe*

Nesse cenário foram reunidos ataques na classe *probe* em um único *dataset*. A razão para isso se deve ao fato de que ataques nessa classe geralmente são executados em sequência. Isso posto, para essa classe foram realizados os seguintes ataques: varredura de IP, varredura de porta e escaneamento de porta. Consequentemente, a quantidade de detectores para o qual o sistema obteve melhor eficácia foram 80, 100, 150 e 200. Para todos esses cenários a adição da técnica de correlação melhorou a eficácia na detecção, no entanto as taxas de detecção e falso negativo não sofreram variação para nenhum deles.

A seguir são apresentadas tabelas e gráficos de barra exibindo a eficácia obtida:

Tabela 72 – MAIS-IDS - *Probe*

80 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	93.265%	0.533%	11.567%	88.433%
STD	0.063%	0.232%	0.09%	0.09%
Max	93.426%	0.67%	11.826%	88.478%
Min	93.206%	0.056%	11.522%	88.174%

Tabela 73 – Abordagem Proposta - *Probe*

80 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	93.341%	0.36%	11.567%	88.433%
STD	0.032%	0.145%	0.09%	0.09%
Max	93.451%	0.446%	11.826%	88.478%
Min	93.304%	0.056%	11.522%	88.174%

Tabela 74 – MAIS-IDS - *Probe*

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	93.294%	0.393%	11.624%	88.376%
STD	0.073%	0.262%	0.142%	0.142%
Max	93.426%	0.67%	11.913%	88.478%
Min	93.231%	0.056%	11.522%	88.087%

Tabela 75 – Abordagem Proposta - *Probe*

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	93.348%	0.271%	11.624%	88.376%
STD	0.054%	0.161%	0.142%	0.142%
Max	93.451%	0.446%	11.913%	88.478%
Min	93.255%	0.056%	11.522%	88.087%

Tabela 76 – MAIS-IDS - *Probe*

150 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	93.305%	0.315%	11.665%	88.335%
STD	0.07%	0.243%	0.128%	0.128%
Max	93.426%	0.67%	11.913%	88.478%
Min	93.206%	0.112%	11.522%	88.087%

Tabela 77 – Abordagem Proposta - *Probe*

150 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	93.343%	0.229%	11.665%	88.335%
STD	0.05%	0.151%	0.128%	0.128%
Max	93.426%	0.446%	11.913%	88.478%
Min	93.255%	0.112%	11.522%	88.087%

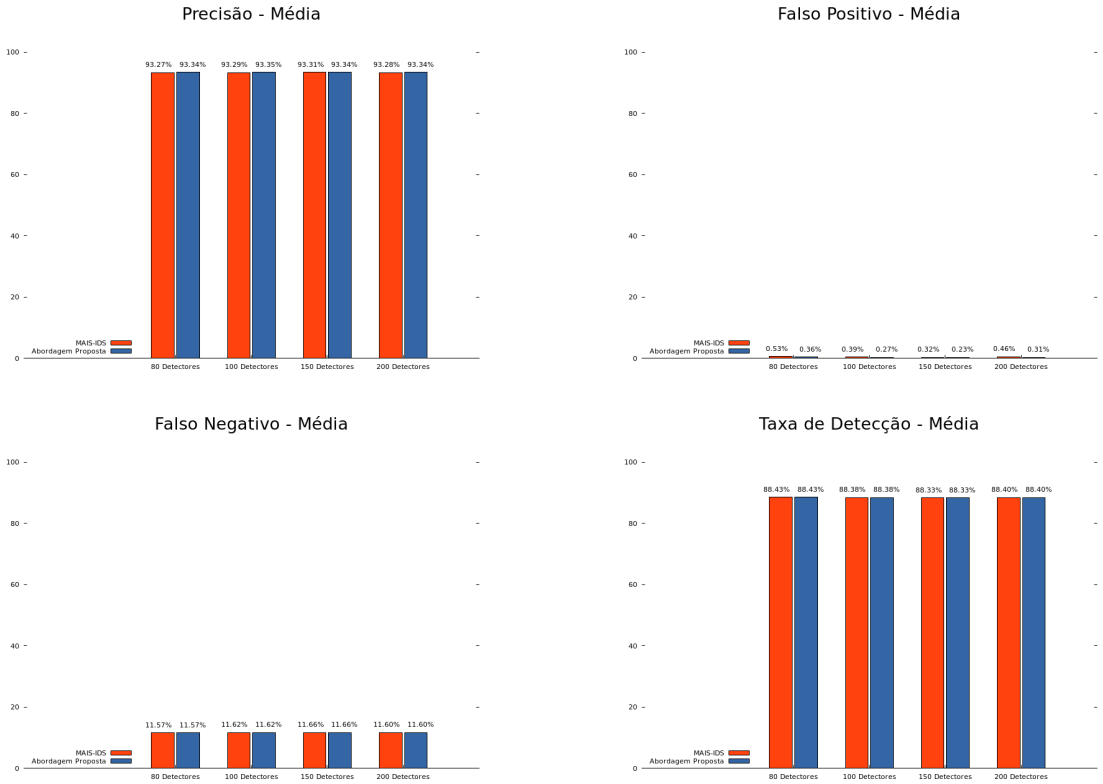


Figura 56 – *Probe* - Média

Figura 57 – *Probe* - STD

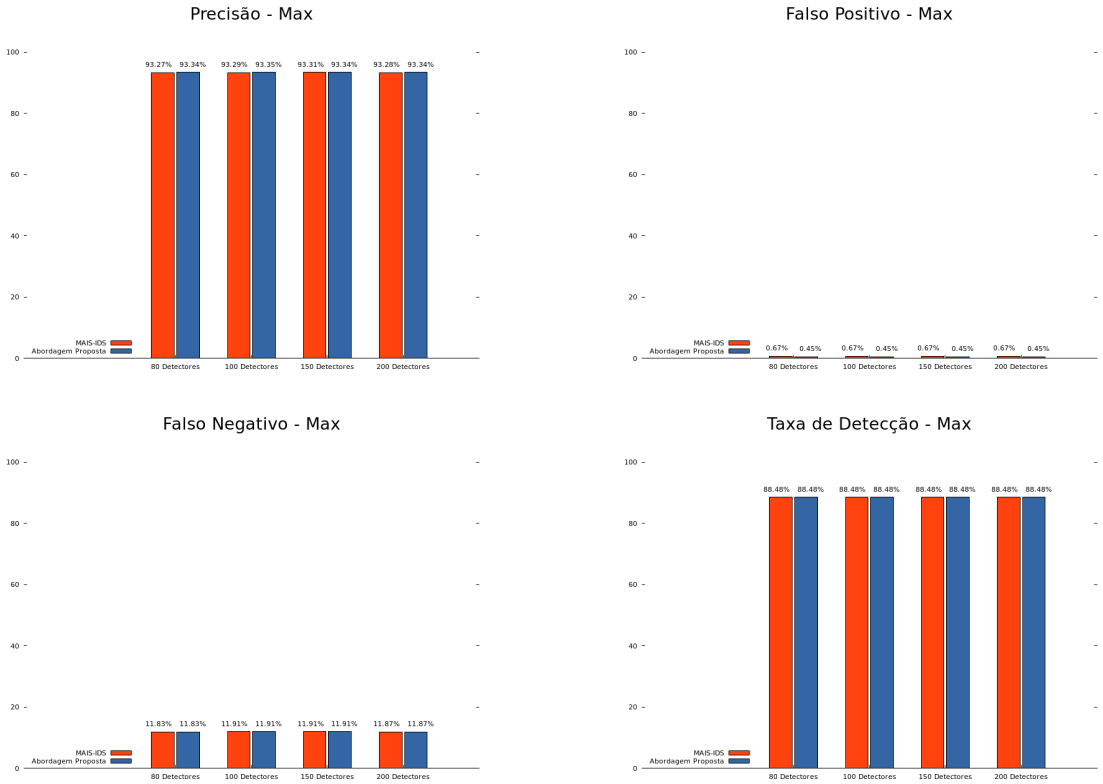


Figura 58 – *Probe* - Max

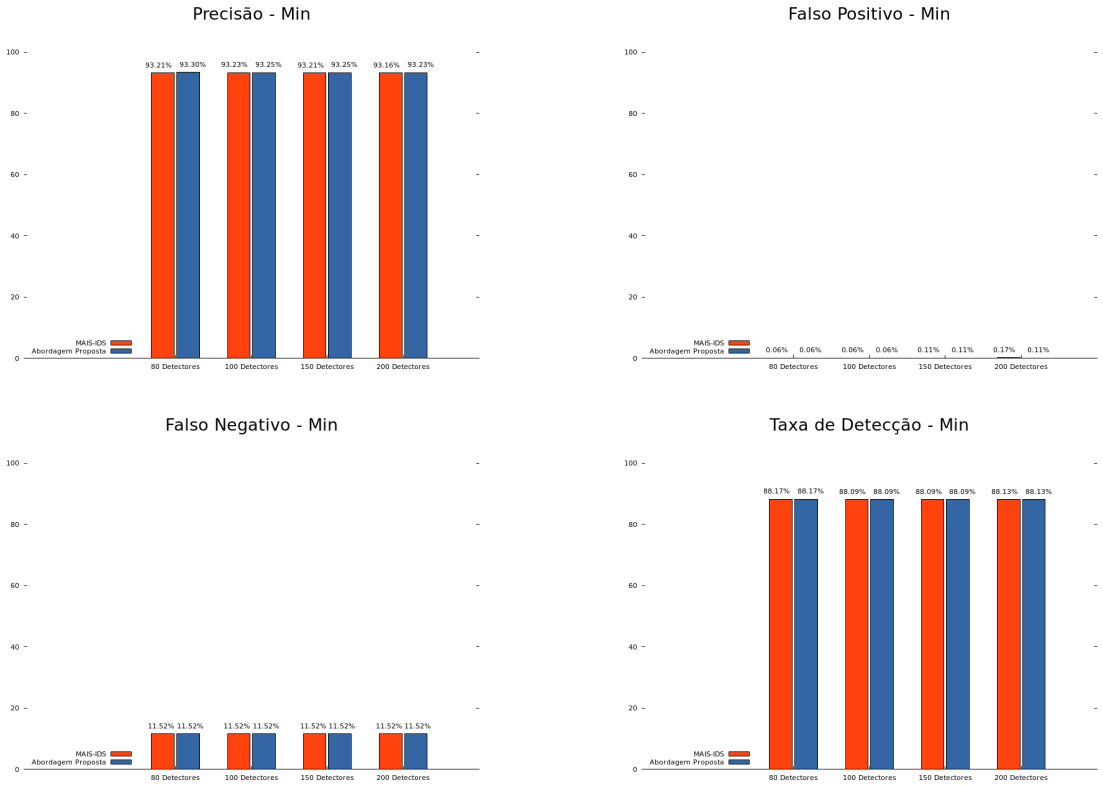


Figura 59 – *Probe* - Min

Tabela 78 – MAIS-IDS - *Probe*

200 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	93.278%	0.455%	11.604%	88.396%
STD	0.078%	0.232%	0.106%	0.106%
Max	93.426%	0.67%	11.87%	88.478%
Min	93.157%	0.167%	11.522%	88.130%

Tabela 79 – Abordagem Proposta - *Probe*

200 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	93.343%	0.307%	11.604%	88.396%
STD	0.054%	0.153%	0.106%	0.106%
Max	93.451%	0.446%	11.87%	88.478%
Min	93.231%	0.112%	11.522%	88.130%

Para a quantidade de 80 detectores a precisão do sistema sofreu um acréscimo na média seguida de uma redução no seu desvio padrão. Ocorreu também uma variação entre os valores máximo e mínimo atingidos, onde os dois sofreram um pequeno acréscimo e diminuíram a diferença entre eles, o que pode ter contribuído para a redução no desvio em relação a média. Já em relação a quantidade de alertas falsos, houve uma redução na média dessa métrica de 32.5%, seguida de outra redução no STD. Uma possível causa para a queda do desvio se deve a redução do valor máximo atingido para alertas falsos, enquanto o valor mínimo permaneceu inalterado.

O sistema continuou apresentando melhorias na eficácia para 100 detectores. Além disso, o comportamento das métricas em relação a adição de correlação de alertas se assemelhou ao cenário anterior. Contudo, para 150 detectores, apesar do sistema apresentar um comportamento semelhante aos dois cenários anteriores, uma diferença se deve ao fato do valor máximo da precisão permanecer constante após a adição da correlação. Já para 200 detectores, a diferença identificada está associada a variação dos valores máximo e mínimo da métrica de falso positivo, onde esses dois valores sofreram redução.

Para ataques nessa categoria, o sistema apresentou melhora em sua eficácia para todas as quantidades de detectores para o qual foi testado. No entanto não houve uma grande variação na diferença entre os valores mínimo e máximo da precisão, resultando em um desvio padrão baixo para essa métrica. No geral, o desvio padrão foi baixo para todas as outras métricas, onde devido a uma quantidade reduzida de alarmes falsos, a adição da técnica de correlação resultou em um pequeno aumento na precisão.

7.2 Segundo Ambiente experimental

Nesse ambiente, a abordagem proposta foi comparada com o MAIS-IDS para as classes de ataque R2L e U2R.

7.2.1 Remoto para Local

Ataques do tipo R2L costumam gerar pouco tráfego de rede, e por essa razão os ataques realizados nessa classe foram reunidos em um único *dataset*. A eficácia da detecção apresentou melhores resultados para 300, 400, 500 e 750 detectores. Em todas as quantidades no qual o sistema foi testado, a técnica de correlação apresentou melhoras na eficácia.

A seguir são apresentadas tabelas e gráficos de barra que demonstram os resultados obtidos para essa classe de ataque.

Tabela 80 – MAIS-IDS - R2L

300 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	74.84%	26.947%	23.898%	76.102%
STD	9.894%	12.302%	18.278%	18.278%
Max	86.254%	50.933%	79.322%	90.485%
Min	41.402%	5.533%	9.515%	20.678%

Tabela 81 – Abordagem Proposta - R2L

300 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	77.276%	1.387%	37.8%	62.2%
STD	12.545%	5.938%	18.626%	18.626%
Max	81.728%	27.267%	100%	68.818%
Min	37.952%	0.0%	31.182%	0.0%

Tabela 82 – MAIS-IDS - R2L

400 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	76.866%	25.957%	21.14%	78.86%
STD	6.688%	11.926%	15.925%	15.925%
Max	87.441%	46%	78.804%	92.275%
Min	50.897%	7.067%	7.725%	21.196%

Tabela 83 – Abordagem Proposta - R2L

400 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	78.751%	1.687%	35.071%	64.929%
STD	8.866%	5.695%	14.864%	14.864%
Max	81.507%	26%	99.859%	68.629%
Min	41.016%	0.0%	31.371%	0.141%

Tabela 84 – MAIS-IDS - R2L

500 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	80.384%	23.91%	16.644%	83.356%
STD	5.568%	13.107%	8.931%	8.931%
Max	90.284%	61.067%	29.204%	90.674%
Min	69.224%	10.2%	9.326%	70.796%

Tabela 85 – Abordagem Proposta - R2L

500 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	80.864%	1.54%	31.569%	68.431%
STD	2.003%	4.899%	0.242%	0.242%
Max	81.617%	22.333%	31.936%	68.959%
Min	72.371%	0.0%	31.041%	68.064%

Tabela 86 – MAIS-IDS - R2L

750 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	76.669%	24.66%	22.393%	77.607%
STD	7.583%	9.621%	15.127%	15.127%
Max	87.248%	41.533%	75.695%	91.145%
Min	52.36%	7.933%	8.855%	24.305%

Tabela 87 – Abordagem Proposta - R2L

750 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	79.413%	0.37%	34.87%	65.13%
STD	8.517%	0.758%	14.457%	14.457%
Max	81.673%	3.267%	97.88%	68.959%
Min	42.313%	0.0%	31.041%	2.12%

Referente a precisão, o sistema apresentou uma média de 77.726% contra 74.84%, apresentando uma melhora considerável para essa métrica quando adicionada a técnica de correlação, considerando a quantidade de 300 detectores. Também houve um crescimento no desvio padrão, apesar da redução na diferença entre os valores máximo e mínimo obtidos. Já a média de falso positivo reduziu em 94.9%, seguida de uma queda significativa no valor de STD, acrescido de uma considerável redução na diferença entre os valores máximo e mínimo alcançados, onde ambos sofreram redução. No entanto a média de falso negativo sofreu um acréscimo significativo, passando de 23.898% para 37.8%, apesar de um desprezível crescimento no desvio padrão. Além disso, a taxa de falso negativo atingiu um valor máximo de 100%, onde houve também um crescimento em seu valor mínimo. A média da taxa de detecção também sofreu uma considerável redução, com um pequeno aumento no STD, e reduções consideráveis nos seus valores máximo e mínimo, onde o valor mínimo alcançado para essa métrica atingiu 0.0%.

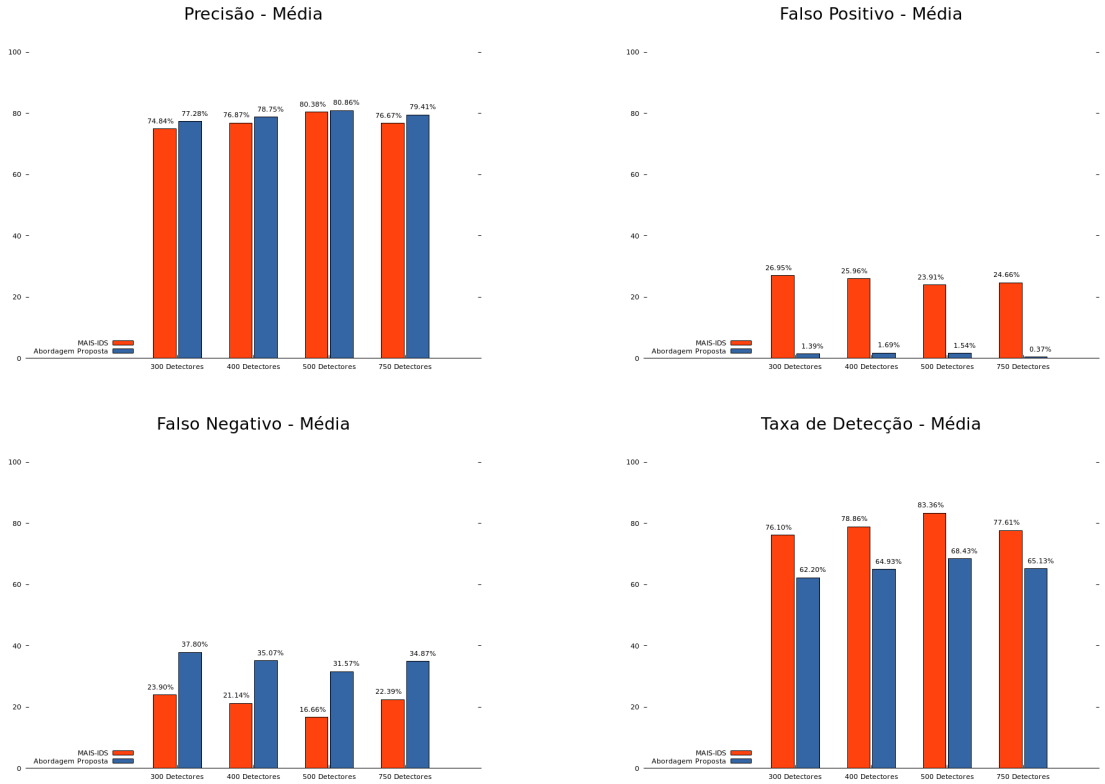


Figura 60 – R2L - Média

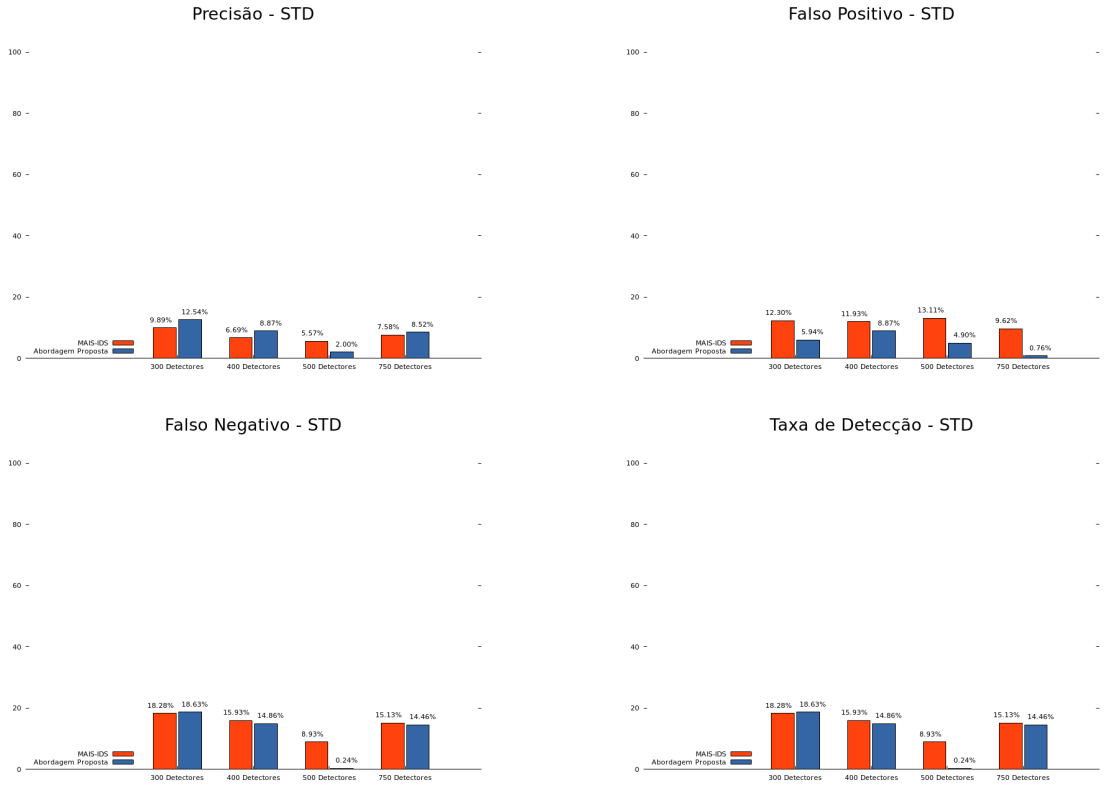


Figura 61 – R2L - STD

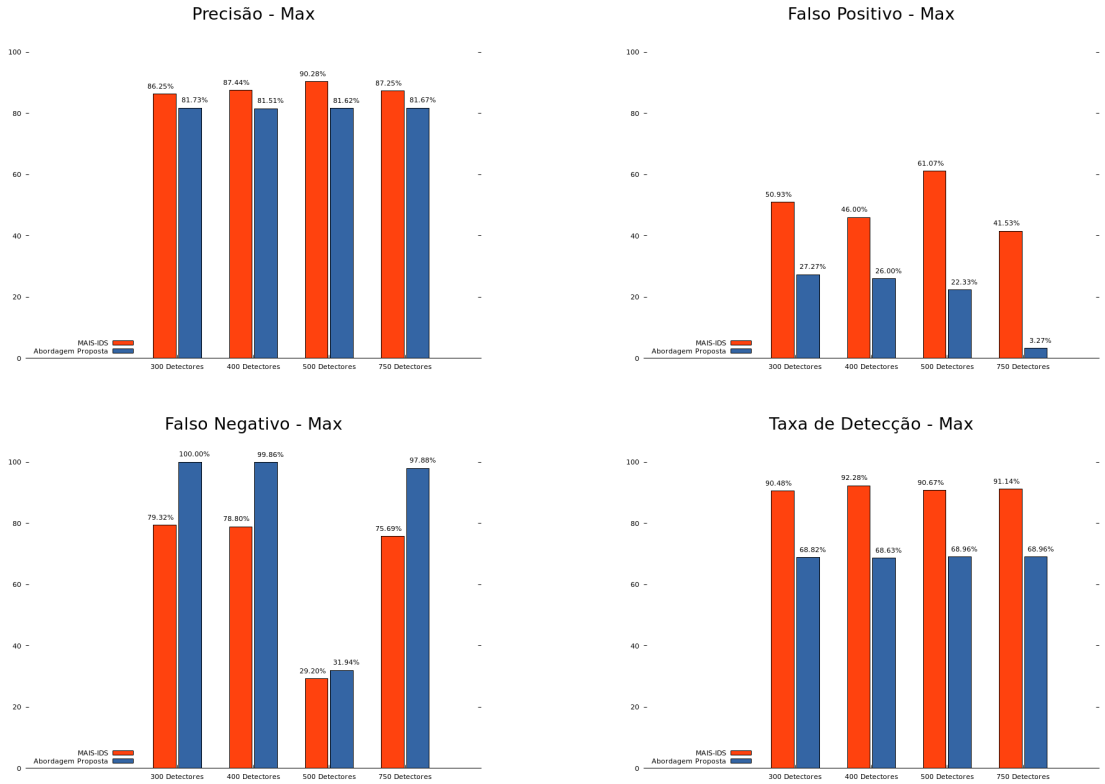


Figura 62 – R2L - Max

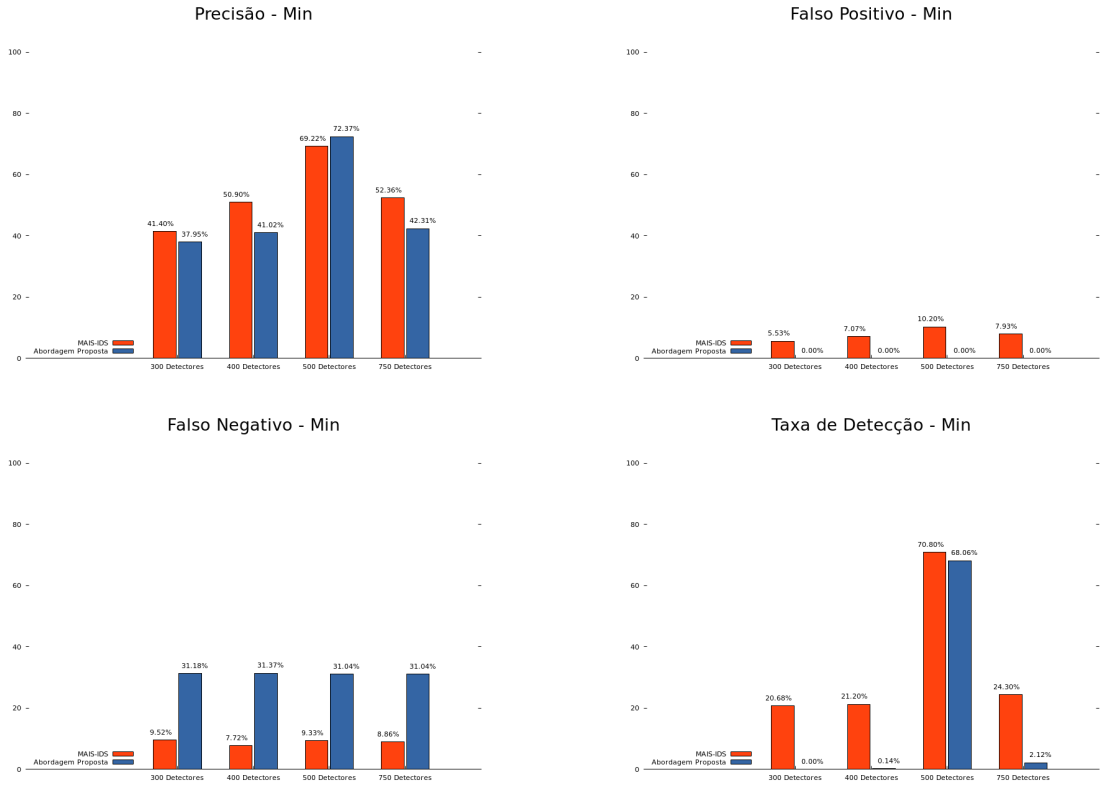


Figura 63 – R2L - Min

Para 400 detectores o sistema apresentou resultados semelhantes ao cenário anterior, quando adicionada a técnica de correlação. A diferença ocorreu no desvio padrão da métrica falso negativo, onde para 400 detectores esse valor reduziu. Além disso, com o acréscimo de detectores, o sistema aumentou sua precisão se comparado com suas versões correspondentes, ou seja, versões na presença e ausência de correlação. Já para 500 detectores, a média da precisão permaneceu aumentando. Para esse cenário houve algumas diferenças comparado aos anteriores. A primeira se deve ao crescimento do valor mínimo atingido na precisão devido a adição de correlação. A segunda se deve a métrica de falso negativo, onde o seu desvio padrão reduziu consideravelmente. A razão para isso está associada a uma grande redução na diferença dos valores máximo e mínimo atingidos. Enquanto em cenários anteriores a adição de correlação aumentava o valor máximo para cerca de 100%, nesse cenário os valores máximo e mínimo correspondem a 31.936% e 31.041% respectivamente. Além disso, a taxa de detecção alcançou seu menor valor em 68.064% enquanto nos cenários anteriores essa redução era próxima de 0.0%. Quando testado com 750 detectores, os resultados apresentados se assemelharam aos dois primeiros, onde a diferença se encontra na redução do valor máximo de alertas falsos alcançados. O sistema obteve uma taxa de 3.267%, enquanto para outras quantidades de detectores esse valor variou entre 22.333% e 27.267%.

Para esse cenário, a média da precisão apresentou resultados inferiores a ataques das categorias anteriores (*Probe*, e DoS), uma vez que algoritmos de aprendizagem de máquina apresentam eficácia pobre para as classes U2R e R2L (JEYA; RAVICHANDRAN; RAVICHANDRAN, 2012). Além disso, diferente dos resultados nas classes anteriores, a adição do método de correlação de alertas além de ter reduzido consideravelmente a taxa de detecção, aumentou a quantidade de alarmes falso negativos. Apesar disso, a adição da técnica de correlação melhorou a eficácia do sistema, uma vez que a precisão aumentou para todos os valores de detectores para o qual o sistema foi testado.

7.2.2 Usuário para Super Usuário

Assim como na classe anterior, ataques U2R geram uma quantidade reduzida de tráfego de rede, e por essa razão o sistema foi testado através de um único *dataset* contendo ataques nessa categoria. As quantidades de detectores para os quais o sistema apresentou melhor eficácia foram 50, 100, 200 e 300. Para todas as quantidades testadas a adição de correlação de alertas melhorou a eficácia da detecção. As tabelas e gráficos de barra apresentam os resultados obtidos para essa classe de ataques.

No primeiro teste foram utilizados 50 detectores, onde o sistema apresentou um aumento na média da precisão. Esse aumento seguiu de uma considerável redução no desvio padrão, variando de 4.413% para 0.06% com a adição da técnica de correlação. A razão para um desvio padrão baixo se deve a pequena diferença entre os valores máximo e mínimo atingidos pela precisão. Já a métrica de falso positivo permaneceu em 0.0% e não variou no decorrer de 20

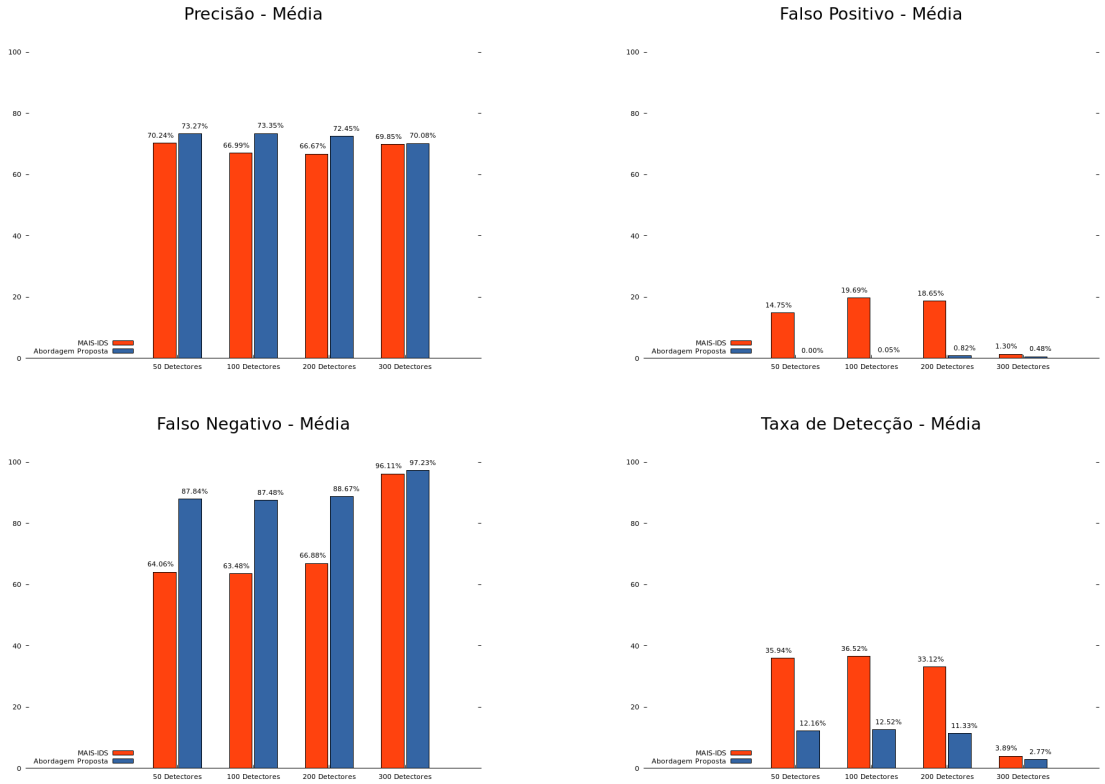


Figura 64 – U2R - Média

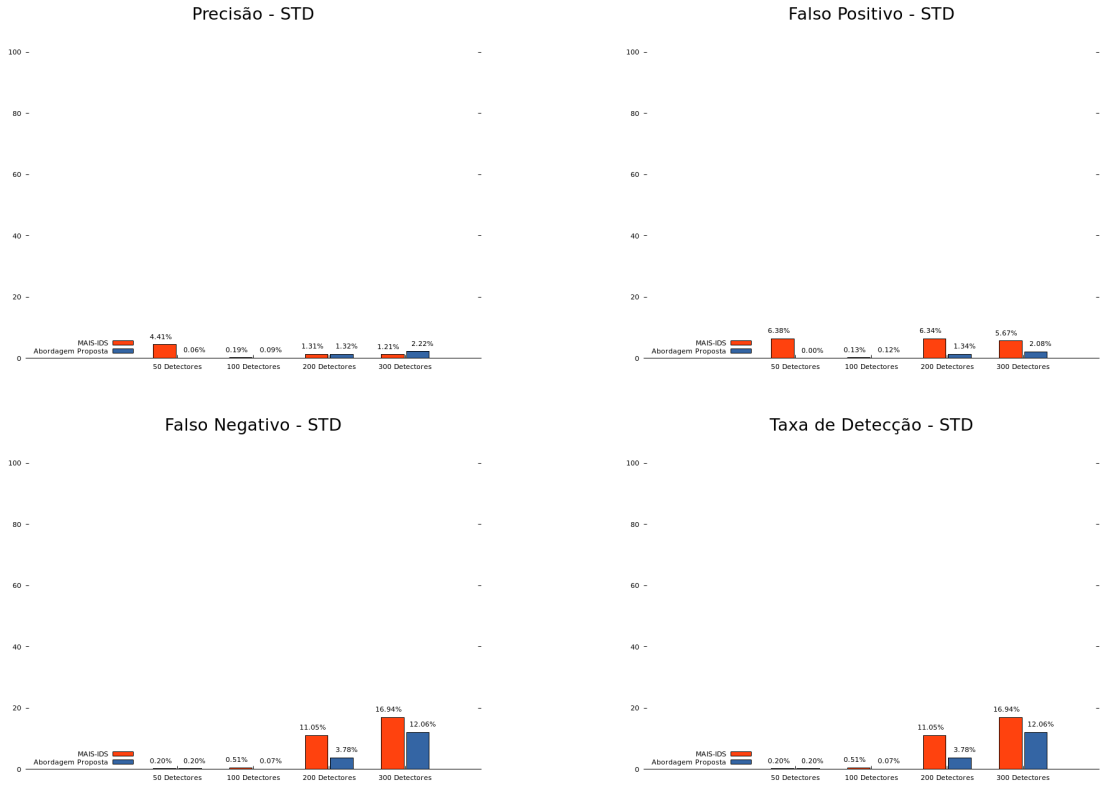


Figura 65 – U2R - STD

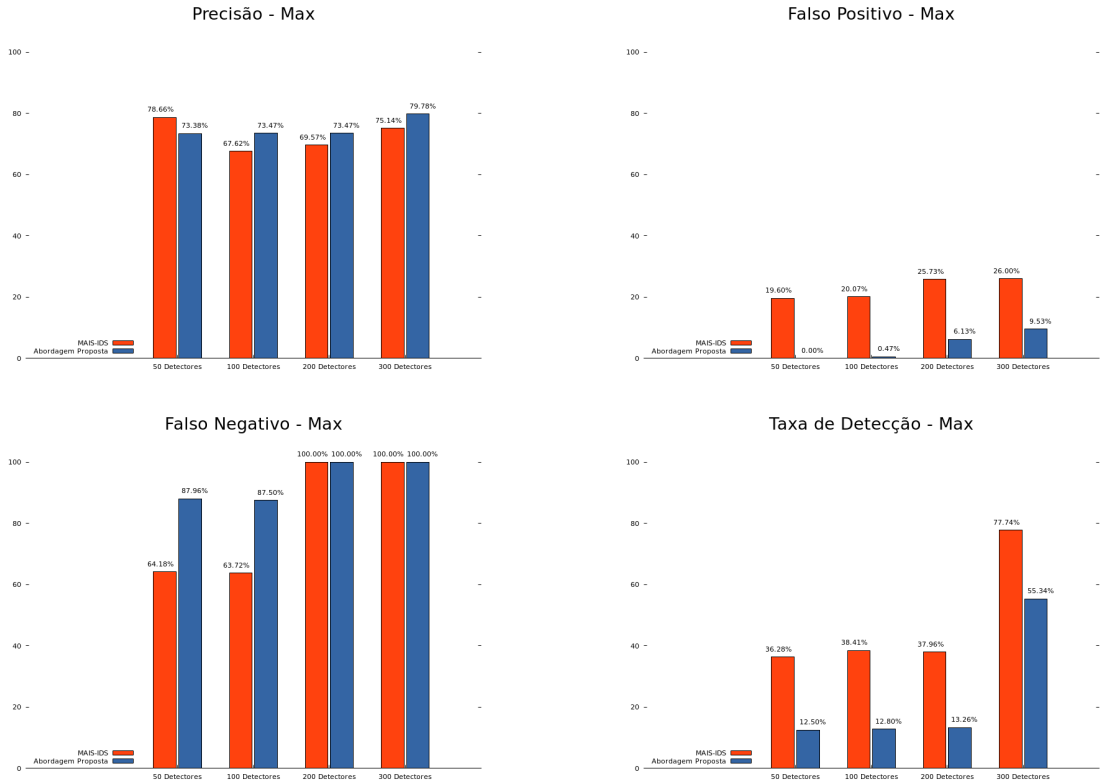


Figura 66 – U2R - Max

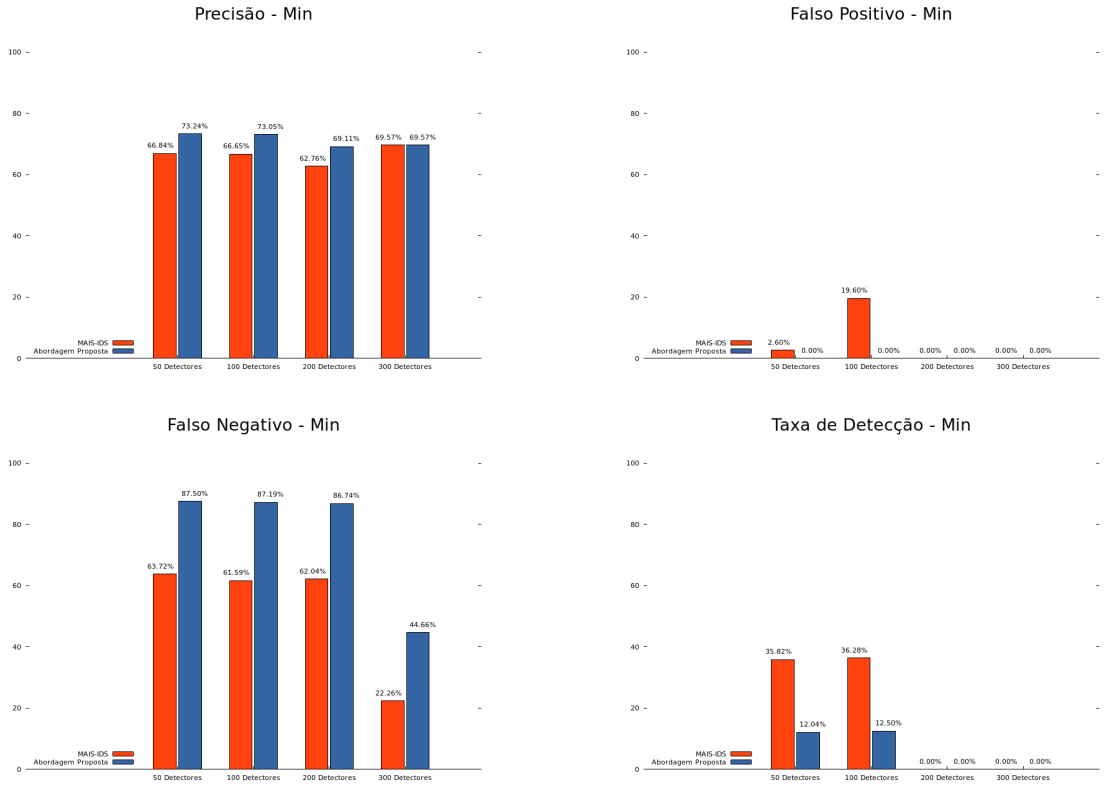


Figura 67 – U2R - Min

Tabela 88 – MAIS-IDS - U2R

50 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	70.244%	14.753%	64.062%	35.938%
STD	4.413%	6.38%	0.198%	0.198%
Max	78.664%	19.6%	64.177%	36.28%
Min	66.837%	2.6%	63.72%	35.823%

Tabela 89 – Abordagem Proposta - U2R

50 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	73.272%	0%	87.843%	12.157%
STD	0.06%	0%	0.198%	0.198%
Max	73.377%	0%	87.957%	12.5%
Min	73.237%	0%	87.5%	12.043%

Tabela 90 – MAIS-IDS - U2R

100 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	66.99%	19.687%	63.476%	36.524%
STD	0.192%	0.127%	0.511%	0.511%
Max	67.625%	20.067%	63.72%	38.415%
Min	66.651%	19.6%	61.585%	36.28%

Tabela 91 – Abordagem Proposta - U2R

100 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	73.346%	0.053%	87.477%	12.523%
STD	0.086%	0.115%	0.073%	0.073%
Max	73.469%	0.467%	87.5%	12.805%
Min	73.052%	0%	87.195%	12.5%

Tabela 92 – MAIS-IDS - U2R

200 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	66.674%	18.65%	66.883%	33.117%
STD	1.307%	6.343%	11.049%	11.049%
Max	69.573%	25.733%	100%	37.957%
Min	62.755%	0%	62.043%	0%

Tabela 93 – Abordagem Proposta - U2R

200 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	72.451%	0.817%	88.674%	11.326%
STD	1.319%	1.339%	3.779%	3.779%
Max	73.469%	6.133%	100%	13.262%
Min	69.109%	0%	86.738%	0%

Tabela 94 – MAIS-IDS - U2R

300 Detectores				
MAIS-IDS	***Acc***	***FP***	***FN***	***DR***
Média	69.852%	1.3%	96.113%	3.887%
STD	1.213%	5.667%	16.944%	16.944%
Max	75.139%	26.0%	100%	77.744%
Min	69.573%	0%	22.256%	0%

Tabela 95 – Abordagem Proposta - U2R

300 Detectores				
IDS-ACG	***Acc***	***FP***	***FN***	***DR***
Média	70.083%	0.477%	97.233%	2.767%
STD	2.224%	2.078%	12.06%	12.06%
Max	79.777%	9.533%	100%	55.335%
Min	69.573%	0%	44.665%	0%

execuções. Contudo, houve um crescimento na média da taxa de falso negativo, seguido de um desvio padrão que permaneceu constante. Enquanto houve um aumento nos valores máximo e mínimo atingidos, porém a diferença entre eles não se alterou, o que contribuiu para que o desvio padrão não sofresse variação. A média da taxa de detecção reduziu consideravelmente, passando de 35.938% para 12.157%. No entanto não houve grande variação em relação a média dessa métrica no decorrer de 20 execuções, isso se evidencia ao se analisar o valor do STD.

Para 100 detectores houve uma maior diferença na média da precisão, passando de 66.99% para 73.346%. No entanto, para o sistema sem a técnica de correlação, os valores máximo e mínimo diminuíram, quando comparado ao cenário anterior também na ausência de correlação. Para o cenário atual, ao se adicionar a técnica em questão, esses valores aumentaram, seguidos de uma redução no desvio padrão, referente a precisão. Já a métrica de falso positivo não obteve uma redução de 100% na média de alertas falsos, como anteriormente. As métricas de falso negativo e taxa de detecção sofreram variações semelhantes as sofridas previamente, quando adicionado o método de correlação. Referente a quantidades maiores de detectores, a adição da técnica aumentou a diferença entre os valores máximo e mínimo da precisão, em relação ao seu uso em cenários anteriores. Isso resultou em um acréscimo do desvio padrão

devido ao uso de correlação. Para 300 detectores, a sua adição aumentou a taxa de detecção para o valor máximo de 55.335%, enquanto para cenários anteriores esse valor permaneceu ao redor de 12.5%.

Para essa categoria, o sistema proposto apresentou os piores resultados. A partir da análise das tabelas, é possível perceber que a taxa de detecção é muito baixa em relação aos grupos de ataques anteriores, assim como as taxas de alarmes falso negativos são bem elevadas. Além disso, a precisão também apresentou resultados pobres, uma vez que sistemas de detecção que se utilizam de algoritmos de aprendizagem de máquina apresentam resultados não satisfatórios para essa classe de ataques (JEYA; RAVICHANDRAN; RAVICHANDRAN, 2012). Apesar desses fatores, a adição da técnica de correlação de alertas melhorou a eficácia para todas as quantidades de detectores para o qual o sistema foi testado, uma vez que houve um aumento considerável na média da precisão. Enquanto a razão para uma taxa de detecção muito baixa, se deve ao fato da ferramenta MulVAL não ter detectado vulnerabilidades em uma das máquinas virtuais, impedindo o mapeamento de alertas associados a essa VM para o grafo de ataque.

7.3 Discussão

Nessa seção são discutidos os resultados para todos os ataques realizados contra o modelo de segurança proposto, de uma forma geral. Serão descritas as vantagens e desvantagens, assim como os melhores e piores cenários para essa abordagem. Em seguida, serão discutidas algumas dificuldades encontradas durante as etapas de implementação, e de realização de testes. Dificuldades associadas as ferramentas utilizadas, como por exemplo o *software* MulVAL (OU; GOVINDAVAJHALA; APPEL, 2005), ao preparo dos cenários de testes, assim como a fase de programação. Onde na última, foi necessário obter um entendimento pleno de todas as técnicas utilizadas para implementação do modelo de segurança proposto. Tarefa na qual foram encontradas dificuldades devido a forma como as informações são apresentadas por certas pesquisas, principalmente pelos artigos (CHUNG et al., 2013), (SERESHT; AZMI, 2014).

7.3.1 Resultados Gerais

A partir dos resultados apresentados pôde-se concluir que a adição da técnica de correlação, para todos os cenários de ataque testados, não reduziu a eficácia do sistema de detecção. A razão para isso se deve ao fato de que a precisão do sistema não diminuiu para nenhum cenário de testes. Apesar disso, para certos cenários houve uma redução na taxa de detecção, e aumento na taxa de alarmes falso negativos. Principalmente para as classes de ataque U2R e R2L, que apresentaram maior degradação na taxa de deteção, devido a um aumento substancial na quantidade de falso negativo. No entanto, um fator importante a ser notado, refere-se a realização dos ataques para compor os *datasets* dessa dissertação, pois para isso foram utilizadas ferramentas mais atuais, referentes a uma versão atual do Kali Linux. Consequentemente, os ataques de

DDoS HTTP e DoS Land não apresentaram uma eficácia satisfatória, considerando que nesses cenários a abordagem proposta foi testada contra apenas um tipo de ataque. Já o mesmo não ocorre para a maior parte das pesquisas voltadas a sistemas de detecção dentro da abordagem imunológica, pois eles são testados por meio de bases como NSL-KDD (University of New Brunswick, 2018), em que os ataques foram realizados por meio de ferramentas mais antigas para compor os *datasets* (MOUSTAFA; SLAY, 2015). Esse processo pode ter favorecido as suas eficácias em comparação ao modelo de segurança proposto.

Na classe DoS, o ataque para o qual o sistema apresentou maior diferença na eficácia refere-se ao DoS PoD, pois de todos os ataques de negação de serviço realizados, esse apresentou maior taxa de alertas falsos. Consequentemente, com a adição da técnica de correlação, foi possível reduzir bastante essa taxa, alcançando uma redução de até 98.2% para 80 detectores. Enquanto para DoS *Teardrop*, houve uma redução de 100% nessa taxa, ela ocorreu justamente para a quantidade de detectores para o qual a adição de correlação apresentou maior eficácia, no caso 80. Para os tipos de negação de serviço HTTP e *Slowloris*, a adição da técnica melhorou a eficácia do sistema a partir da quantidade de 200. Em relação ao DDoS HTTP, a maior redução de alarmes falsos alcançou 13.7% para 400 detectores, enquanto para DDoS *Slowloris*, alcançou 36.2% para 300. Já para os ataques de DoS Land e DoS *Smurf*, a adição do método de correlação reduziu a taxa de falso positivo apenas para a quantidade de 20 detectores, onde essa redução alcançou os respectivos valores de 96.8% e 37.2 %. Enquanto para o tipo de negação de serviço DoS TCP *Flood*, houve redução de 6.8% a partir do valor de 400. Já considerando todos os ataques de DoS realizados em um único *dataset*, houve redução de 6.2% apenas para 50 deles. Ataques de reconhecimento apresentaram uma leve melhora na precisão, onde a maior redução na taxa de alarmes falsos alcançou 32.5% para o valor de 80, enquanto as classes R2L e U2R obtiveram 98.5% para a quantidade de 300, e 100% para 50, respectivamente. No entanto para a classe U2R, a taxa de detecção apresentou valores muito reduzidos, ao redor de 12%.

De acordo com os resultados apresentados, percebe-se que a técnica de correlação apresentou uma maior redução na taxa de alertas falsos para as categorias R2L e U2R. Já para as classes DoS e *probe* a redução foi bem menor, no entanto se tratam justamente das classes na qual o sistema obteve melhor eficácia de detecção. Comparando-se os resultados apresentados com o do artigo (MOUSSAID; TOUMANARI; AZHARI, 2017), para certas classes de ataque a redução na taxa de alertas falsos é bem maior para a abordagem proposta. O artigo em questão, utiliza um IDS baseado em assinatura em conjunto com a mesma técnica de correlação utilizada pelo modelo de segurança proposto, onde o sistema é testado contra ataques multi-níveis e DDoS. No entanto o método de correlação é utilizado em conjunto com uma técnica de clusterização, e apresenta uma redução de 64% comparada a sistemas de detecção de assinatura sem o auxilio dessa técnica. Já a abordagem proposta apresentou resultados em que essa redução obteve valores máximos de até 100%, chegando a valores mínimos de 6.2%. Sendo assim, é importante identificar em que cenários de ataque seria melhor empregar o modelo de segurança proposto.

Por se tratar de uma técnica cuja função está associada a redução da quantidade de alertas falsos gerados, a correlação de alerta surtirá maior efeito contra ataques que geram maiores números de falso positivos. No entanto, em cenários sujeitos a muitos ataques, em que boa parte deles se encontram na classe U2R, onde o sistema precisa manter uma alta taxa de detecção, não seria recomendado a utilização do método de correlação. Pois pelos resultados evidenciou-se que a adição de correlação reduziu bastante a taxa de detecção. Já para ataques na classe R2L, embora a queda na taxa de detecção não seja crítica como para a classe anterior, seria interessante evitar a utilização desse método em cenários com alta incidência de ataques, onde a maior parte deles se encontram nessas duas categorias. Já para a categoria DoS, não existem ressalvas em relação a utilização desse método, pois o aspecto negativo verificado se deve a uma desprezível queda na taxa de detecção devido a um pequeno aumento na quantidade de falso negativo. Já para ataques do tipo *probe*, a abordagem proposta melhorou a eficácia da detecção sem nenhum custo adicional.

Apesar de existirem alguns cenários não favoráveis a utilização da abordagem proposta, a utilização da técnica de correlação se apresentou viável, pois para nenhum ataque realizado a sua adição resultou em eficácia inferior ao do sistema de detecção sem o auxilio de correlação. Inclusive, houve cenários em que a taxa de redução de alertas falsos foi bem superior ao do artigo (MOUSSAID; TOUMANARI; AZHARI, 2017). Além disso, por não existir a necessidade de um alerta gerado se encontrar associado a alguma vulnerabilidade na rede, bastando apenas que o mapeamento ocorra por meio de sua origem e destino, o modelo proposto é capaz de mitigar diversos cenários de ataque em seus estágios iniciais. A razão para isso se deve a sua capacidade de detectar ataques que não se encontram associados a nenhuma vulnerabilidade relacionada ao sistema CVSS (MELL; SCARFONE; ROMANOSKY, 2006), como no caso de muitos ataques na categoria de reconhecimento ou *probe*. Pois ataques desse tipo viabilizam futuras ameaças como negação de serviço, ou acesso não autorizado como usuário ou super usuário a uma determinada máquina. Adicionando essa vantagem a capacidade que o modelo de segurança proposto possui de selecionar a contra medida ótima de acordo com a métrica *return of investiment*, ela contribui para aumentar o estado de segurança da rede. Esse estado se encontra associado a probabilidade condicional de cada vulnerabilidade que corresponde a probabilidade que ela possui de ser comprometida.

7.3.2 Análise de Desempenho

A pesquisa realizada nessa dissertação foca na análise da abordagem proposta em relação a sua eficácia na classificação dos dados estatísticos do tráfego de rede em normais ou anômalos. A análise de desempenho referente a métricas como consumo de CPU, ou memória, e tempo de detecção e mitigação de ataques, estão fora do escopo dessa pesquisa. Sendo assim, essa subseção se foca em discutir sobre o desempenho da abodagem de segurança proposta.

Em trabalhos anteriores (CHUNG et al., 2013) (XING et al., 2013) (MOUSSAID;

TOUMANARI; AZHARI, 2017), as arquiteturas de segurança propostas analisam o tráfego de rede a nível de pacote. Uma vantagem dessa abordagem se deve a análise em tempo real. Em um eventual cenário de intrusão, a ameaça poderia ser detectada mais rapidamente em relação a abordagem proposta. Uma vez que no modelo de segurança desenvolvido a análise ocorre a nível de dados estatísticos do tráfego de rede, o sistema de detecção precisa esperar um determinado tempo para a passagem de pacotes em cada VM, para compor cada padrão estatístico. No entanto, esse tipo de análise possibilita que abordagens de segurança escalem melhor no ambiente da nuvem, uma vez que a sobrecarga na rede é menor. Para exemplificar, a arquitetura (CHUNG et al., 2013) apresenta problemas na sobrecarga da rede relacionados ao consumo de cpu, atraso de comunicação entre máquinas virtuais e na taxa de sucesso de pacotes analisados. Problemas que também são enfrentados pelos trabalhos (XING et al., 2013) (MOUSSAID; TOUMANARI; AZHARI, 2017), devido a utilização de sistemas de detecção com o mesmo tipo de análise de tráfego para o ambiente da nuvem.

Já o desempenho do MAIS-IDS foi medido em um trabalho anterior (SERESHT; AZMI, 2014), onde foi utilizado um servidor HP DL380 G, com dois núcleos de processamento e 348 GB de memória. O resultado obtido para esse sistema foi um consumo de 0.6% de CPU e 200-300 MB de memória.

O atraso introduzido na prevenção de ataques devido aos algoritmos de correlação e de seleção de contramedidas pode ser analisado por meio da ordem de complexidade de cada um deles. O primeiro possui uma ordem de complexidade de $O(|V|+|A|)$, em que "V"se refere a quantidade de vulnerabilidades presentes na rede, enquanto "A"está associado a quantidade de alertas gerados. Já para o algoritmo de seleção, a ordem de complexidade é de $O(|V|x|CM|)$, onde "CM"se trata de uma constante, referente ao número de contramedidas SDN que podem ser executadas, e "V"se trata da quantidade de vulnerabilidades presentes na rede. Já que a ordem de complexidade apresentada por cada um dos algoritmos é linear, nenhum deles introduz um considerável atraso ao processo de prevenção.

No que se refere ao processo de geração de grafos de ataque, a ferramenta MulVAL é utilizada (OU; GOVINDAVAJHALA; APPEL, 2005). Essa ferramenta escala bem para redes de grande porte, uma vez que a analise dos dados coletados para geração do grafo pode ser realizada em um curto intervalo de tempo (OU; GOVINDAVAJHALA; APPEL, 2005). No entanto, o escaneamento da rede para detecção de vulnerabilidades se trata de um processo lento. Apesar disso, as etapas de escaneamento e geração do grafo de ataque são necessárias apenas quando um usuário instala aplicativos, pois novas vulnerabilidades podem ser introduzidas no ambiente da nuvem.

7.3.3 Dificuldades Encontradas

Em relação a implementação dos algoritmos utilizados, a primeira dificuldade encontrada está relacionada ao trabalho de (SERESHT; AZMI, 2014). Esse artigo apresenta um sistema de

detecção baseado em agentes distribuídos. Um dos fatores que dificultou sua implementação se deve ao processo de sincronização das instâncias do IDS. Pois cada instância é composta por agentes que precisam trabalhar de forma sincronizada para que o sistema não trave durante a etapa de detecção. Esse processo não foi detalhado pelo artigo (SERESHT; AZMI, 2014), e por essa razão houve dificuldades durante sua implementação. Já em relação a técnica de correlação, implementada pelo algoritmo apresentado em (CHUNG et al., 2013), que utiliza a abordagem de (ROSCHKE; CHENG; MEINEL, 2011), não foram apresentados exemplos práticos de como esse método auxiliaria no processo de detecção em um cenário real de ataque. O mesmo se deve ao algoritmo de seleção de contra-medidas utilizado por (CHUNG et al., 2013). Outro problema se deve a definição de grafo de ataque desenvolvida por (CHUNG et al., 2013). Uma vez que não são apresentados detalhes sobre o processo de conversão do grafo gerado pelo *software* MulVAL para o formato proposto por esse artigo.

Outra dificuldade encontrada está associada a análise do tráfego de rede. Onde primeiro foi realizada uma pesquisa de como analisar esses dados em tempo real no formato NSL-KDD (University of New Brunswick, 2018). No entanto não foram encontradas soluções práticas para análise em tempo real nessa etapa. Portanto, foi utilizada uma abordagem (PERONA, 2013) que converte o tráfego de rede para um formato similar ao KDDCup99 (University of California, Irvine, 2018). Essa conversão ocorre por meio do tráfego capturado por outra ferramenta utilizada pelo *Linux*, conhecida como *tcpdump*. Essa escolha não prejudicou a pesquisa realizada, pois as métricas utilizadas para medir a eficácia da detecção não exigem que o sistema realize testes em tempo real.

Outro problema está associado ao *software* MulVAL, pois foram encontradas dificuldades em sua utilização. Todas a versões encontradas não estavam funcionando corretamente, inclusive a versão obtida diretamente do *site* original. Acredita-se que a razão para isso se deve a falta de suporte para essa ferramenta, uma vez que ela não sofria atualizações frequentemente. Posto que o *software* MulVAL trabalha em conjunto com outros programas, ele precisa ser constantemente atualizado.

Por último, se encontram dificuldades relacionadas a realização dos ataques, assim como do preparo dos ambientes experimentais. Uma vez que a técnica de correlação utilizada por esse trabalho necessita de dados de vulnerabilidade e topologia da rede, foi necessário gerar o próprio conjunto de dados do tráfego de rede para que fosse possível testar o sistema. Posto que *datasets* conhecidos (KDD, NSL-KDD) não fornecem nenhum tipo de informação sobre o ambiente no qual eles foram gerados que possibilite a construção de um grafo de ataque. Além disso, para a maior parte dos ataques testados, era necessário adicionar uma certa vulnerabilidade para uma determinada VM. Para isso, era necessário primeiro instalar alguma aplicação que tornasse a VM vulnerável a alguma ameaça. No entanto, nem sempre o aplicativo instalado funcionava corretamente, impedindo o comprometimento da VM. Outras vezes, apesar da versão vulnerável do aplicativo se encontrar em plena execução, o ataque falhava em comprometer a máquina

virtual.

8 Conclusões e Trabalhos Futuros

Esse trabalho apresentou uma nova abordagem de segurança para combater ataques que exploram vulnerabilidades. Ameaças desse tipo costumam ocorrer tanto em redes convencionais, como no ambiente da nuvem, e se encontram divididos nas categorias DoS, U2R, R2L e reconhecimento. Essa abordagem é baseada na detecção de anomalias, na qual as técnicas imunológicas funcionam em conjunto com grafos de ataque e correlação de alertas. Assim, foi possível desenvolver um modelo de segurança baseado em agentes distribuídos, onde cada alerta gerado pode ser correlacionado e mapeado para um grafo de ataque. Como resultado, a taxa de alarmes falsos pôde ser reduzida por meio da técnica de correlação.

O sistema de segurança desenvolvido foi testado remotamente através do *Amazon Web Service*. A adição da técnica de correlação baseada em grafos de ataque apresentou melhores resultados para as quatro categorias de ataques estudadas. De uma forma geral, o sistema proposto obteve melhor precisão, com uma quantidade reduzida de alarmes falsos, onde a exceção se deve aos grupos U2R e R2L, onde também houve um aumento na taxa de falso negativo, resultando na degradação da taxa de detecção. A razão para a redução de alertas falsos se deve ao processo de mapeamento e correlação de cada alerta gerado. Uma vez que esse trabalho pressupõe que os ataques ocorrem apenas de fora da rede, o primeiro alerta gerado a partir de um novo cenário de ataque deve ter sua origem na Internet, caso contrário, ele será considerado falso. Caso o cenário de ataque não seja novo, o alerta gerado deverá ser mapeado para um vértice de conjunção no grafo de ataque. Em seguida, o novo alerta precisa ser correlacionado com outros gerados anteriormente no ACG. Esse processo resulta na diminuição da taxa de alarmes falsos, especialmente para conjuntos de dados com um alto número de padrões normais, resultando no aumento da precisão do sistema, e na sua eficácia geral.

Essa pesquisa apresentou um modelo de segurança capaz de detectar e mitigar ataques em seus estágios iniciais. A razão para isso se deve a capacidade do sistema em detectar ataques de reconhecimento que não se encontram associados a nenhuma vulnerabilidade presente em bancos de dados conhecidos, como (The MITRE Corporation, 2018), ou (National Institute of Standards and Technology, 2018). Isso é possível porque os alertas gerados precisam apenas de informações dos endereços de origem e destino para mapear o alerta para uma máquina vulnerável encontrada pelo *software* MulVAL (OU; GOVINDAVAJHALA; APPEL, 2005). Enquanto sistemas de detecção baseados em assinatura, utilizam um modo de mapeamento no qual são necessárias informações do endereço de origem, destino, assim como do tipo de alerta. O tipo de alerta geralmente se encontra associado a uma determinada vulnerabilidade. Sendo assim, para esses sistemas é necessário que a assinatura do ataque se encontre associada a uma determinada vulnerabilidade para que o processo de mapeamento possa ocorrer. No entanto, se o modo de mapeamento for o mesmo da abordagem proposta, ainda assim, o IDS apresentará as

mesmas limitações relacionadas a uma abordagem de assinatura.

8.1 Contribuições

Essa pesquisa procura fornecer um melhor entendimento de como os conceitos de correlação de alertas baseado em grafos de ataque, redes SDN, e sistemas de detecção que utilizam a abordagem imunológica, podem contribuir para um ambiente mais seguro que visa combater ataques que exploram vulnerabilidades da infraestrutura na nuvem. Sendo assim, as contribuições desse trabalho foram as seguintes:

- Identificar as técnicas mais propícias para NIDS baseados na abordagem imunológica no ambiente da nuvem, uma vez que a seleção e execução de contra-medidas depende do desempenho da técnica de detecção escolhida. A partir de técnicas e algoritmos que representem o estado da arte na abordagem imunológica será possível selecionar e executar contra-medidas mais apropriadas para cada cenário de intrusão;

- Desenvolver uma abordagem de segurança que explique e forneça um melhor entendimento de como os conceitos de correlação de alertas, redes SDN, e imunologia podem ser utilizados em conjunto para proteger os serviços de infraestrutura da nuvem. Uma vez que no modelo proposto o grafo de ataque pode auxiliar na seleção de contra-medidas e no processo de detecção, é importante entender como ele pode influenciar sistemas de detecção e de prevenção para que se possa tirar maior proveito desse recurso. Grafos de ataque representam os possíveis caminhos que um atacante poderia seguir para explorar vulnerabilidades, e pode ser utilizado em conjunto com técnicas de correlação de alertas, cuja função é de mapear e correlacionar os alertas gerados.

- Apresentar os resultados do modelo proposto e compará-lo ao mesmo sistema sem a utilização de correlação de alertas. Dessa forma é possível identificar o quanto essa técnica pode melhorar a eficácia na detecção. A eficácia é medida para cada ataque dentro das categorias de negação de serviço, remoto para local, usuário para super usuário e *probe*.

- Apresentar um modelo de segurança que pode servir de suporte tanto para arquiteturas de segurança na nuvem, como para redes convencionais. Onde uma de suas vantagens está relacionada à redução de alertas falsos por meio de uma técnica de correlação de alertas. Enquanto outra vantagem se deve a utilização de um sistema de detecção baseado em anomalia. Pois ainda que surja uma nova ameaça associada a uma vulnerabilidade ainda não identificada, ou caso surja uma nova ameaça associada a mesma vulnerabilidade, esse sistema ainda é capaz de detectá-la. Por último, existe a possibilidade de mitigar um ataque reconfigurando a rede por meio da tecnologia SDN, tornando esse ambiente mais seguro.

8.2 Trabalhos Futuros

Para trabalhos futuros, pretende-se apresentar uma arquitetura que suporte o modelo de segurança proposto para redes convencionais, assim como para o ambiente da nuvem. No entanto, antes de apresentar uma nova arquitetura, pretende-se testar a abordagem proposta em cenários mais realistas e complexos. Esses cenários apresentariam um maior número de máquinas virtuais e maior diversidade de serviços. A partir de ambientes mais realistas, a abordagem proposta poderia ser testada em relação a métricas como consumo de memória e utilização de CPU. Pretende-se também pesquisar ferramentas que permitam uma melhor comunicação entres diferentes instâncias do IDS, permitindo que o modelo proposto se adapte melhor em ambientes mais complexos. Também podem ser pesquisadas diferentes abordagens de seleção de contramedidas baseadas em grafos de ataque e redes programáveis.

Além disso, serão realizados testes para coletar o estado de segurança da rede. Esse estado está relacionado à probabilidade condicional que cada vulnerabilidade possui de ser comprometida. Dessa forma, contramedidas SDN podem ser executadas para mitigar uma ameaça, para que depois se analise a variação de probabilidade referente a cada vulnerabilidade após o ataque. Essa variação ocorre devido a reconfiguração da rede, e geração de um novo grafo de ataque, onde novas probabilidades condicionais são calculadas. Já em relação a detecção, pretende-se testar o sistema com *datasets* mais atuais, que apresentem diferentes atributos do tráfego de rede que possam melhorar a eficácia da detecção. Também serão realizados testes no que se refere ao desempenho da abordagem proposta, considerando o tempo que o sistema gasta desde o processo de detecção até a mitigação de ataques.

Referências

AHMAD, A.; IDRIS, N. B.; KAMA, M. N. Cloudids: Cloud intrusion detection model inspired by dendritic cell mechanism. *International Journal of Communication Networks and Information Security (IJCNIS)*, v. 9, n. 1, 2017. Citado 3 vezes nas páginas 18, 48 e 66.

AKYAZI, U.; UYAR, Ş. A hybrid multiobjective evolutionary algorithm for anomaly intrusion detection. In: *Distributed Computing and Artificial Intelligence*. [S.l.]: Springer, 2010. p. 509–516. Citado na página 48.

ALI, K.; AIB, I.; BOUTABA, R. P2p-ais: a p2p artificial immune systems architecture for detecting ddos flooding attacks. In: IEEE. *Information Infrastructure Symposium, 2009. GIIS'09. Global*. [S.l.], 2009. p. 1–4. Citado na página 48.

Amazon Web Service. *Amazon Web Service*. 2018. <https://aws.amazon.com/pt/>, acessado em 7 de Outubro . Disponível em: <https://aws.amazon.com/pt/>. Citado na página 72.

AMBEDKAR, C.; BABU, V. K. Detection of probe attacks using machine learning techniques. *International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)*, v. 2, n. 3, p. 25–29, 2015. Citado 5 vezes nas páginas 17, 26, 27, 28 e 75.

ANDERSON, J. P. et al. *Computer security threat monitoring and surveillance*. [S.l.], 1980. Citado na página 14.

AYARA, M. et al. Negative selection: How to generate detectors. In: CANTERBURY, UK:[SN]. *Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS)*. [S.l.], 2002. v. 1, p. 89–98. Citado na página 35.

BERSINI, H.; VARELA, F. Hints for adaptive problem solving gleaned from immune networks. *Parallel problem solving from nature*, Springer, p. 343–354, 1991. Citado na página 32.

BHARDWAJ, S.; JAIN, L.; JAIN, S. Cloud computing: A study of infrastructure as a service (iaas). *International Journal of engineering and information Technology*, v. 2, n. 1, p. 60–63, 2010. Citado na página 25.

BOX, G. E. et al. *Time series analysis: forecasting and control*. [S.l.]: John Wiley & Sons, 2015. Citado na página 56.

BROWNLEE, J. *Clever algorithms: nature-inspired programming recipes*. [S.l.]: Jason Brownlee, 2011. Citado na página 72.

BUILDING Blocks of SDN Network, Acessado em 10 de maio de 2017 em https://nutanshinde.wordpress.com/2016/01/31/building-blocks-of-sdn-network/. In: . [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 4 e 41.

BURAGOHAIN, C.; MEDHI, N. Flowtrapp: An sdn based architecture for ddos attack detection and mitigation in data centers. In: IEEE. *Signal Processing and Integrated Networks (SPIN), 2016 3rd International Conference on*. [S.l.], 2016. p. 519–524. Citado na página 28.

BURCH, H.; CHESWICK, B. Tracing anonymous packets to their approximate source. In: *LISA*. [S.l.: s.n.], 2000. p. 319–327. Citado na página 59.

BUYYA, R. et al. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, Elsevier, v. 25, n. 6, p. 599–616, 2009. Citado na página 23.

CARPINTERIA Rural FRIEDRICH, Acessado em 15 de maio de 2017 em http://carpinteriarural.com/index.php/discount/25727/. In: . [S.l.: s.n.], 2017. Citado 2 vezes nas páginas 4 e 26.

CHEN, M.-H.; CHANG, P.-C.; WU, J.-L. A population-based incremental learning approach with artificial immune system for network intrusion detection. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 51, p. 171–181, 2016. Citado 2 vezes nas páginas 50 e 65.

CHEN, X.-F.; YU, S.-Z. Cipa: A collaborative intrusion prevention architecture for programmable network and sdn. *Computers & Security*, Elsevier, v. 58, p. 1–19, 2016. Citado 3 vezes nas páginas 4, 60 e 61.

CHOU, T.-S. Security threats on cloud computing vulnerabilities. *International Journal of Computer Science & Information Technology*, Academy & Industry Research Collaboration Center (AIRCC), v. 5, n. 3, p. 79, 2013. Citado na página 29.

CHUNG, C.-J. et al. Nice: Network intrusion detection and countermeasure selection in virtual network systems. *IEEE transactions on dependable and secure computing*, IEEE, v. 10, n. 4, p. 198–211, 2013. Citado 22 vezes nas páginas 4, 16, 18, 19, 20, 26, 39, 40, 61, 62, 64, 65, 66, 67, 68, 69, 70, 73, 120, 122, 123 e 124.

COOKE, D. E.; HUNT, J. E. Recognising promoter sequences using an artificial immune system. In: *ISMB*. [S.l.: s.n.], 1995. p. 89–97. Citado na página 33.

DAMN Vulnerable Web Application. 2018. <http://xylos.wikidot.com/home/>, acessado em 15 de Setembro. Disponível em: <http://xylos.wikidot.com/home/>. Citado na página 73.

DEBAR, H.; CURRY, D.; FEINSTEIN, B. *The intrusion detection message exchange format (IDMEF)*. [S.l.], 2007. Citado na página 37.

DENNING, D. E. An intrusion-detection model. *IEEE Transactions on software engineering*, IEEE, n. 2, p. 222–232, 1987. Citado na página 14.

DESVENDANDO os modelos de serviços em Cloud, Acessado em 20 de abril de 2017 em https://angelopublio.com.br/modelos-servicos-cloud-saas-paas-iaas/. In: . [S.l.: s.n.], 2017. Citado 3 vezes nas páginas 15, 24 e 25.

D'HAESELEER, P.; FORREST, S.; HELMAN, P. An immunological approach to change detection: Algorithms, analysis and implications. In: IEEE. *Security and Privacy, 1996. Proceedings., 1996 IEEE Symposium on*. [S.l.], 1996. p. 110–119. Citado na página 34.

ELHAJ, M. M.; HAMRAWI, H.; SULIMAN, M. M. A multi-layer network defense system using artificial immune system. In: IEEE. *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*. [S.l.], 2013. p. 232–236. Citado 2 vezes nas páginas 48 e 66.

ESTEVEZ-TAPIADOR, J. M.; GARCIA-TEODORO, P.; DIAZ-VERDEJO, J. E. Stochastic protocol modeling for anomaly based network intrusion detection. In: IEEE. *Information Assurance, 2003. IWIAS 2003. Proceedings. First IEEE International Workshop on*. [S.l.], 2003. p. 3–12. Citado na página 31.

FARID, D. et al. Adaptive network intrusion detection learning: attribute selection and classification. In: *International Conference on computer systems Engineering (ICCSE 2009)*. [S.l.: s.n.], 2009. p. TH60000. Citado na página 76.

FLOODLIGHT. 2018. <http://www.projectfloodlight.org/floodligh>, acessado em 15 de Setembro. Disponível em: <http://www.projectfloodlight.org/floodligh>. Citado 2 vezes nas páginas 53 e 55.

FORREST, S. et al. Self-nonself discrimination in a computer. In: IEEE. *Research in Security and Privacy, 1994. Proceedings., 1994 IEEE Computer Society Symposium on*. [S.l.], 1994. p. 202–212. Citado 2 vezes nas páginas 4 e 34.

FREUND, Y.; SCHAPIRE, R. E. et al. Experiments with a new boosting algorithm. In: CITESEER. *Icml*. [S.l.], 1996. v. 96, p. 148–156. Citado na página 56.

FRIGAULT, M.; WANG, L. Measuring network security using bayesian network-based attack graphs. In: IEEE. *Annual IEEE International Computer Software and Applications Conference*. [S.l.], 2008. p. 698–703. Citado 2 vezes nas páginas 39 e 70.

GARCIA-TEODORO, P. et al. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, Elsevier, v. 28, n. 1, p. 18–28, 2009. Citado 3 vezes nas páginas 4, 30 e 31.

GIOTIS, K. et al. Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments. *Computer Networks*, Elsevier, v. 62, p. 122–136, 2014. Citado na página 57.

GOEL, S.; GANGOLLY, J. S. On decision support for distributed systems protection: A perspective based on the human immune response system and epidemiology. *International Journal of Information Management*, Elsevier, v. 27, n. 4, p. 266–278, 2007. Citado na página 47.

GOGOI, P. et al. Packet and flow based network intrusion dataset. In: SPRINGER. *International Conference on Contemporary Computing*. [S.l.], 2012. p. 322–334. Citado na página 77.

GOLD, N. et al. Understanding service-oriented software. *IEEE software*, IEEE, v. 21, n. 2, p. 71–77, 2004. Citado na página 24.

GONZÁLEZ, F. A.; DASGUPTA, D. Anomaly detection using real-valued negative selection. *Genetic Programming and Evolvable Machines*, Springer, v. 4, n. 4, p. 383–403, 2003. Citado na página 35.

GREENSMITH, J.; AICKELIN, U.; CAYZER, S. Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In: SPRINGER. *International Conference on Artificial Immune Systems*. [S.l.], 2005. p. 153–167. Citado na página 35.

GUO, F.; CHIUEH, T.-c. Sequence number-based mac address spoof detection. In: SPRINGER. *International Workshop on Recent Advances in Intrusion Detection*. [S.l.], 2005. p. 309–329. Citado na página 59.

HA, T. et al. Suspicious traffic sampling for intrusion detection in software-defined networks. *Computer Networks*, Elsevier, v. 109, p. 172–182, 2016. Citado na página 52.

HA, T. et al. Suspicious flow forwarding for multiple intrusion detection systems on software-defined networks. *IEEE Network*, IEEE, v. 30, n. 6, p. 22–27, 2016. Citado na página 54.

HAAG, C. R. et al. An artificial immune system-inspired multiobjective evolutionary algorithm with application to the detection of distributed computer network intrusions. In: *Artificial Immune Systems*. [S.l.]: Springer, 2007. p. 420–435. Citado na página 48.

HALVAIEE, N. S.; AKBARI, M. K. A novel model for credit card fraud detection using artificial immune systems. *Applied Soft Computing*, Elsevier, v. 24, p. 40–49, 2014. Citado na página 33.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011. Citado na página 53.

HARMER, P. K. et al. An artificial immune system architecture for computer security applications. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 3, p. 252–280, 2002. Citado na página 47.

HOMER, J. et al. Aggregating vulnerability metrics in enterprise networks using attack graphs. *Journal of Computer Security*, IOS Press, v. 21, n. 4, p. 561–597, 2013. Citado na página 38.

HOSSEINPOUR, F. et al. Design of a new distributed model for intrusion detection system based on artificial immune system. In: IEEE. *Advanced Information Management and Service (IMS), 2010 6th International Conference on*. [S.l.], 2010. p. 378–383. Citado na página 48.

HOW to Configure and Run Xplico. 2018. <http://xylos.wikidot.com/howto-linux-runxplico/>, acessado em 15 de Setembro. Disponível em: <http://xylos.wikidot.com/howto-linux-runxplico/>. Citado na página 73.

HU, Y.-C.; JAKOBSSON, M.; PERRIG, A. Efficient constructions for one-way hash chains. In: SPRINGER. *International Conference on Applied Cryptography and Network Security*. [S.l.], 2005. p. 423–441. Citado na página 59.

HUANG, N.-F. et al. An openflow-based collaborative intrusion prevention system for cloud networking. In: IEEE. *Communication Software and Networks (ICCSN), 2015 IEEE International Conference on*. [S.l.], 2015. p. 85–92. Citado 2 vezes nas páginas 42 e 52.

HUBBALLI, N.; SURYANARAYANAN, V. False alarm minimization techniques in signature-based intrusion detection systems: A survey. *Computer Communications*, Elsevier, v. 49, p. 1–17, 2014. Citado 5 vezes nas páginas 4, 16, 36, 37 e 38.

HUBBARD, D.; SUTTON, M. et al. Top threats to cloud computing v1. 0. *Cloud Security Alliance*, p. 1–14, 2010. Citado na página 17.

HUNT, J. E.; COOKE, D. E. Learning using an artificial immune system. *Journal of network and computer applications*, Elsevier, v. 19, n. 2, p. 189–212, 1996. Citado na página 33.

IERACE, N.; URRUTIA, C.; BASSETT, R. Intrusion prevention systems. *Ubiquity*, ACM, v. 2005, n. June, p. 2–2, 2005. Citado 3 vezes nas páginas 29, 30 e 32.

IGBE, O.; DARWISH, I.; SAADAWI, T. Distributed network intrusion detection systems: An artificial immune system approach. In: IEEE. *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on*. [S.l.], 2016. p. 101–106. Citado 3 vezes nas páginas 49, 65 e 66.

Imperva. *IMPERVA INCAPSULA*. 2018. <https://www.incapsula.com/>, last accessed on July 8. Disponível em: <https://www.incapsula.com>. Citado 4 vezes nas páginas 28, 74, 75 e 105.

INDULSKA, M.; ORLOWSKA, M. E. Gravity based spatial clustering. In: ACM. *Proceedings of the 10th ACM international symposium on Advances in geographic information systems*. [S.l.], 2002. p. 125–130. Citado na página 54.

ISLAM, T. et al. *A Probabilistic Network Security Metric Based on Attack Graphs*. [S.l.], 2008. Citado na página 39.

JAMJOOM, H.; WILLIAMS, D.; SHARMA, U. Don't call them middleboxes, call them middlepipes. In: ACM. *Proceedings of the third workshop on Hot topics in software defined networking*. [S.l.], 2014. p. 19–24. Citado na página 41.

JENSEN, F. V. *An introduction to Bayesian networks*. [S.l.]: UCL press London, 1996. v. 210. Citado na página 39.

JEONG, C. et al. Scalable network intrusion detection on virtual sdn environment. In: IEEE. *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*. [S.l.], 2014. p. 264–265. Citado na página 54.

JERNE, N. K. Towards a network theory of the immune system. In: *Annales d'immunologie*. [S.l.: s.n.], 1974. v. 125, n. 1-2, p. 373. Citado 2 vezes nas páginas 32 e 36.

JEYA, P. G.; RAVICHANDRAN, M.; RAVICHANDRAN, C. Efficient classifier for r2l and u2r attacks. *International Journal of Computer Applications*, Citeseer, v. 45, n. 21, p. 29, 2012. Citado 4 vezes nas páginas 18, 28, 116 e 120.

JHA, M.; ACHARYA, R. An immune inspired unsupervised intrusion detection system for detection of novel attacks. In: IEEE. *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*. [S.l.], 2016. p. 292–297. Citado 3 vezes nas páginas 50, 65 e 66.

JHA, S.; SHEYNER, O.; WING, J. Two formal analyses of attack graphs. In: IEEE. *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*. [S.l.], 2002. p. 49–63. Citado 2 vezes nas páginas 37 e 38.

JIANG, Y.; CHANG, J. Intrusion prevention system base on immune vaccination. In: IEEE. *Intelligent Computation Technology and Automation, 2009. ICICTA'09. Second International Conference on*. [S.l.], 2009. v. 1, p. 350–353. Citado na página 48.

JIANG, Y.; CHANG, J. Design of network security system base on vaccination. In: IEEE. *Natural Computation (ICNC), 2010 Sixth International Conference on*. [S.l.], 2010. v. 1, p. 224–227. Citado na página 48.

JIANG, Y. et al. A method of in-depth-defense for network security based on immunity principles. In: IEEE. *Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on*. [S.l.], 2009. p. 484–487. Citado na página 48.

KABIRI, P.; GHORBANI, A. A. Research on intrusion detection and response: A survey. *IJ Network Security*, v. 1, n. 2, p. 84–102, 2005. Citado na página 30.

KALLIOLA, A. et al. Flooding ddos mitigation and traffic management with software defined networking. In: IEEE. *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*. [S.l.], 2015. p. 248–254. Citado na página 17.

KARMAKAR, K. K.; VARADHARAJAN, V.; TUPAKULA, U. Mitigating attacks in software defined network (sdn). In: IEEE. *Software Defined Systems (SDS), 2017 Fourth International Conference on*. [S.l.], 2017. p. 112–117. Citado 3 vezes nas páginas 4, 55 e 56.

KAUR, R. et al. Security of software defined networks: Taxonomic modeling, key components and open research area. In: IEEE. *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*. [S.l.], 2016. p. 2832–2839. Citado 2 vezes nas páginas 20 e 21.

KEMMERER, R. A.; VIGNA, G. Intrusion detection: a brief history and overview. *Computer*, IEEE, v. 35, n. 4, p. supl27–supl30, 2002. Citado na página 29.

KIM, J.; BENTLEY, P. The artificial immune model for network intrusion detection. In: *7th European congress on intelligent techniques and soft computing (EUFIT'99)*. [S.l.: s.n.], 1999. v. 158. Citado na página 35.

KREUTZ, D. et al. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, IEEE, v. 103, n. 1, p. 14–76, 2015. Citado na página 41.

KUMAR, G. P.; REDDY, D. K. An agent based intrusion detection system for wireless network with artificial immune system (ais) and negative clone selection. In: IEEE. *Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on*. [S.l.], 2014. p. 429–433. Citado na página 14.

KWON, J. et al. An incrementally deployable anti-spoofing mechanism for software-defined networks. *Computer Communications*, Elsevier, v. 64, p. 1–20, 2015. Citado na página 59.

KWON, K.; AHN, S.; CHUNG, J. W. Network security management using arp spoofing. In: SPRINGER. *International Conference on Computational Science and Its Applications*. [S.l.], 2004. p. 142–149. Citado na página 71.

LAKHINA, A.; CROVELLA, M.; DIOT, C. Mining anomalies using traffic feature distributions. In: ACM. *ACM SIGCOMM Computer Communication Review*. [S.l.], 2005. v. 35, n. 4, p. 217–228. Citado na página 58.

LAWTON, G. Developing software online with platform-as-a-service technology. *Computer*, IEEE, v. 41, n. 6, 2008. Citado na página 24.

LE, A. et al. Flexible network-based intrusion detection and prevention system on software-defined networks. In: IEEE. *Advanced Computing and Applications (ACOMP), 2015 International Conference on*. [S.l.], 2015. p. 106–111. Citado 3 vezes nas páginas 4, 53 e 54.

LI, B.; LIU, P.; LIN, L. A cluster-based intrusion detection framework for monitoring the traffic of cloud environments. In: IEEE. *Cyber Security and Cloud Computing (CSCloud), 2016 IEEE 3rd International Conference on*. [S.l.], 2016. p. 42–45. Citado na página 52.

LIAO, H.-J. et al. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, Elsevier, v. 36, n. 1, p. 16–24, 2013. Citado 2 vezes nas páginas 29 e 30.

LIM, S. et al. A sdn-oriented ddos blocking scheme for botnet-based attacks. In: IEEE. *Ubiquitous and Future Networks (ICUFN), 2014 Sixth International Conf on*. [S.l.], 2014. p. 63–68. Citado 3 vezes nas páginas 17, 28 e 105.

LIPPMANN, R. et al. Analysis and results of the 1999 darpa off-line intrusion detection evaluation. In: SPRINGER. *International Workshop on Recent Advances in Intrusion Detection*. [S.l.], 2000. p. 162–182. Citado 2 vezes nas páginas 17 e 27.

LIU, S. et al. Multi-agent network intrusion active defense model based on immune theory. *Wuhan University Journal of Natural Sciences*, Springer, v. 12, n. 1, p. 167–171, 2007. Citado na página 47.

LOGENTRIES. 2017. <http://www.logentries.com>, acessado em 8 de Setembro. Disponível em: <http://www.logentries.com>. Citado na página 57.

LUTHER, K. et al. A cooperative ais framework for intrusion detection. In: IEEE. *Communications, 2007. ICC'07. IEEE International Conference on*. [S.l.], 2007. p. 1409–1416. Citado 4 vezes nas páginas 18, 20, 21 e 47.

MAHONEY, M. V.; CHAN, P. K. An analysis of the 1999 darpa/lincoln laboratory evaluation data for network anomaly detection. In: SPRINGER. *International Workshop on Recent Advances in Intrusion Detection*. [S.l.], 2003. p. 220–237. Citado na página 77.

MASOUDI, R.; GHAFFARI, A. Software defined networks: A survey. *Journal of Network and Computer Applications*, Elsevier, v. 67, p. 1–25, 2016. Citado 2 vezes nas páginas 40 e 42.

MCHUGH, J. Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC)*, ACM, v. 3, n. 4, p. 262–294, 2000. Citado na página 77.

MCKEOWN, N. et al. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, ACM, v. 38, n. 2, p. 69–74, 2008. Citado na página 63.

MELL, P.; SCARFONE, K.; ROMANOSKY, S. Common vulnerability scoring system. *IEEE Security & Privacy*, IEEE, v. 4, n. 6, 2006. Citado 4 vezes nas páginas 38, 63, 68 e 122.

MOBASHER, B. Web usage mining. In: *Encyclopedia of data warehousing and mining*. [S.l.]: IGI Global, 2005. p. 1216–1220. Citado na página 56.

MOUSSAID, N. E.; TOUMANARI, A.; AZHARI, M. E. Security analysis as software-defined security for sdn environment. In: IEEE. *Software Defined Systems (SDS), 2017 Fourth International Conference on*. [S.l.], 2017. p. 87–92. Citado 9 vezes nas páginas 19, 20, 63, 64, 65, 66, 121, 122 e 123.

MOUSTAFA, N.; SLAY, J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: IEEE. *Military Communications and Information Systems Conference (MilCIS), 2015*. [S.l.], 2015. p. 1–6. Citado na página 121.

National Institute of Standards and Technology. *National vulnerability database*. 2018. <https://nvd.nist.gov/>, acessado em 3 de Julho. Disponível em: <https://nvd.nist.gov/>. Citado 4 vezes nas páginas 68, 72, 75 e 126.

NUNES, B. A. A. et al. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, IEEE, v. 16, n. 3, p. 1617–1634, 2014. Citado na página 41.

OKTAY, U.; SAHINGOZ, O. Attack types and intrusion detection systems in cloud computing. In: *Proceedings of the 6th International Information Security & Cryptology Conference*. [S.l.: s.n.], 2013. p. 71–76. Citado na página 15.

OpenBSD Project. *OpenSSH*. 2018. <https://www.openssh.com>, acessado em 18 de Dezembro. Disponível em: <https://www.openssh.com>. Citado na página 75.

OU, X.; GOVINDAVAJHALA, S.; APPEL, A. W. Mulval: A logic-based network security analyzer. In: BALTIMORE, MD. *USENIX Security Symposium*. [S.l.], 2005. v. 8. Citado 8 vezes nas páginas 4, 39, 70, 72, 76, 120, 123 e 126.

PARTHASARATHY, K. Clonal selection method for immunity-based intrusion detection system. *Location: http://web. umr. deu/tauritzd/courses/cs447/project/, also available from SMG*, 2003. Citado na página 35.

PATEL, A. et al. An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of network and computer applications*, Elsevier, v. 36, n. 1, p. 25–41, 2013. Citado na página 32.

PEARL, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. [S.l.]: Elsevier, 2014. Citado na página 39.

PERELSON, A. S. Immune network theory. *Immunological reviews*, Wiley Online Library, v. 110, n. 1, p. 5–36, 1989. Citado na página 32.

PERONA, I. *GureKddcup database description*. 2013. Citado 2 vezes nas páginas 72 e 124.

PHAAL, P.; PANCHEN, S.; MCKEE, N. *InMon corporation's sFlow: A method for monitoring traffic in switched and routed networks*. [S.l.], 2001. Citado na página 58.

QIAO, P.; SU, J.; SUN, C. Distributed intrusion detection system based on artis. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2005*. [S.l.], 2005. v. 5812, p. 356–364. Citado na página 47.

QIU, X.; ZHANG, K.; REN, Q. Global flow table: A convincing mechanism for security operations in sdn. *Computer Networks*, Elsevier, v. 120, p. 56–70, 2017. Citado na página 53.

QUINLAN, J. R. *C4. 5: programs for machine learning*. [S.l.]: Elsevier, 2014. Citado na página 54.

Radware. *DDoS Warriors*. 2018. <https://security.radware.com/>, last accessed on July 8. Disponível em: <https://security.radware.com>. Citado 2 vezes nas páginas 74 e 75.

ROSCHKE, S.; CHENG, F.; MEINEL, C. A new alert correlation algorithm based on attack graph. In: *Computational intelligence in security for information systems*. [S.l.]: Springer, 2011. p. 58–67. Citado 6 vezes nas páginas 62, 63, 64, 68, 69 e 124.

RUIRUI, Z. et al. The research of network intrusion detection based on danger theory and cloud model. In: SPRINGER. *International Conference on Information and Management Engineering*. [S.l.], 2011. p. 204–211. Citado na página 48.

SABAHI, F. Secure virtualization for cloud environment using hypervisor-based technology. *International Journal of Machine Learning and Computing*, IACSIT Press, v. 2, n. 1, p. 39, 2012. Citado 2 vezes nas páginas 21 e 24.

SADODDIN, R.; GHORBANI, A. Alert correlation survey: framework and techniques. In: ACM. *Proceedings of the 2006 international conference on privacy, security and trust: bridge the gap between PST technologies and business services*. [S.l.], 2006. p. 37. Citado na página 36.

SALAH, S.; MACIÁ-FERNÁNDEZ, G.; DÍAZ-VERDEJO, J. E. A model-based survey of alert correlation techniques. *Computer Networks*, Elsevier, v. 57, n. 5, p. 1289–1317, 2013. Citado na página 36.

SAYEED, M. A.; SAYEED, M. A.; SAXENA, S. Intrusion detection system based on software defined network firewall. In: IEEE. *Next Generation Computing Technologies (NGCT), 2015 1st International Conference on*. [S.l.], 2015. p. 379–382. Citado na página 53.

SEEBER, S.; RODOSEK, G. D. Towards an adaptive and effective ids using openflow. In: SPRINGER. *IFIP International Conference on Autonomous Infrastructure, Management and Security*. [S.l.], 2015. p. 134–139. Citado 5 vezes nas páginas 4, 18, 21, 42 e 51.

SEKAR, R. et al. Specification-based anomaly detection: a new approach for detecting network intrusions. In: ACM. *Proceedings of the 9th ACM conference on Computer and communications security*. [S.l.], 2002. p. 265–274. Citado na página 31.

SERESHT, N. A.; AZMI, R. Mais-ids: A distributed intrusion detection system using multi-agent ais approach. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 35, p. 286–298, 2014. Citado 10 vezes nas páginas 17, 49, 65, 66, 67, 72, 79, 120, 123 e 124.

SHAMSHIRBAND, S. et al. Co-fais: cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks. *Journal of Network and Computer Applications*, Elsevier, v. 42, p. 102–117, 2014. Citado na página 14.

SHANDILYA, V.; SIMMONS, C. B.; SHIVA, S. Use of attack graphs in security systems. *Journal of Computer Networks and Communications*, Hindawi Publishing Corporation, v. 2014, 2014. Citado na página 37.

SHANG, F. et al. Distributed controllers multi-granularity security communication mechanism for software-defined networking. *Computers & Electrical Engineering*, Elsevier, v. 66, p. 388–406, 2018. Citado na página 60.

SHIN, S.; GU, G. Attacking software-defined networks: A first feasibility study. In: ACM. *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. [S.l.], 2013. p. 165–166. Citado 2 vezes nas páginas 15 e 40.

SILVA, G. C.; PALHARES, R. M.; CAMINHAS, W. M. A transitional view of immune inspired techniques for anomaly detection. In: SPRINGER. *International Conference on Intelligent Data Engineering and Automated Learning*. [S.l.], 2012. p. 568–577. Citado 2 vezes nas páginas 4 e 36.

SINIARSKI, B. et al. Real-time monitoring of sdn networks using non-invasive cloud-based logging platforms. In: IEEE. *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016 IEEE 27th Annual International Symposium on*. [S.l.], 2016. p. 1–6. Citado 3 vezes nas páginas 4, 57 e 58.

SISTEMAS de Detecção de Intrusões (IDS – Intrusion Detection Systems) usando unicamente softwares Open Source, Acessado em 15 de abril de 2017 em https://seginfo.com.br/2010/06/21/sistemas-de-deteccao-de-intrusoes-ids-intrusion-detection-systems-usando-unicamente-softwares-open-source/. In: . [S.l.: s.n.], 2010. Citado na página 14.

SOOD, M. et al. A survey on issues of concern in software defined networks. In: IEEE. *Image Information Processing (ICIIP), 2015 Third International Conference on*. [S.l.], 2015. p. 295–300. Citado na página 41.

SOURCE Forge. 2018. <https://sourceforge.net/projects/lamahub/>, acessado em 10 de Setembro. Disponível em: <https://sourceforge.net/projects/lamahub/>. Citado na página 73.

SOUSA, F. R.; MOREIRA, L. O.; MACHADO, J. C. Computação em nuvem: Conceitos, tecnologias, aplicações e desafios. *II Escola Regional de Computação Ceará, Maranhão e Piauí (ERCEMAPI)*, p. 150–175, 2009. Citado na página 25.

SRIHARI, V.; KALPANA, P.; ANITHA, R. Dendritic cell algorithm for preventing spam over ip telephony. In: IEEE. *Informatics, Electronics & Vision (ICIEV), 2014 International Conference on*. [S.l.], 2014. p. 1–6. Citado na página 14.

SU, Z. et al. Cemon: A cost-effective flow monitoring system in software defined networks. *Computer Networks*, Elsevier, v. 92, p. 101–115, 2015. Citado 3 vezes nas páginas 4, 58 e 59.

SUBASHINI, S.; KAVITHA, V. A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, Elsevier, v. 34, n. 1, p. 1–11, 2011. Citado 3 vezes nas páginas 15, 24 e 25.

TAVALLAEE, M. et al. A detailed analysis of the kdd cup 99 data set. In: IEEE. *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*. [S.l.], 2009. p. 1–6. Citado na página 77.

TELNET Protocol Specification. 2018. <https://tools.ietf.org/html/rfc854/>, acessado em 20 de Setembro. Disponível em: <https://tools.ietf.org/html/rfc854/>. Citado na página 73.

Tenable. *Nessus Vulnerability Scanner*. 2018. <https://www.tenable.com/>, acessado em 8 de Agosto. Disponível em: <https://www.tenable.com/>. Citado na página 72.

The MITRE Corporation. *Common Vulnerabilities and Exposures (CVE)*. 2018. <http://cve.mitre.org/>, acessado em 3 de junho. Disponível em: <http://cve.mitre.org/>. Citado 3 vezes nas páginas 72, 75 e 126.

TIMMIS, J. et al. An overview of artificial immune systems. *Computation in Cells and Tissues: Perspectives and Tools for Thought*, Natural Computation Series, Springer, p. 51–86, 2004. Citado 2 vezes nas páginas 33 e 34.

TURNER, M.; BUDGEN, D.; BRERETON, P. Turning software into a service. *Computer*, IEEE, v. 36, n. 10, p. 38–44, 2003. Citado na página 24.

University of California, Irvine. *KDD Cup 1999 Data*. 2018. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html/>, acessado em 7 de Julho. Disponível em: <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>. Citado 3 vezes nas páginas 50, 77 e 124.

University of Minnesota Duluth. *CLASSIFICATION METHODS*. 2018. <http://www.d.umn.edu/~padhy005/Chapter5.html>, acessado em 7 de Julho. Disponível em: <http://www.d.umn.edu/~padhy005/Chapter5.html>. Citado na página 49.

University of New Brunswick. *NSL-KDD dataset*. 2018. <http://www.unb.ca/cic/datasets/nsl.html/>, acessado em 7 de Julho. Disponível em: <http://www.unb.ca/cic/datasets/nsl.html>. Citado 6 vezes nas páginas 49, 50, 72, 76, 121 e 124.

UWAGBOLE, S.; BUCHANAN, W.; FAN, L. Novel tracking of rogue network packets using danger theory approach. *The Institute Ecole Supérieure en Informatique Electronique et Automatique, Laval, France 5-6 July 2012 Edited by*, p. 277, 2012. Citado na página 48.

VASUDEVAN, A.; HARSHINI, E.; SELVAKUMAR, S. Ssenet-2011: a network intrusion detection system dataset and its comparison with kdd cup 99 dataset. In: IEEE. *Internet (AH-ICI), 2011 Second Asian Himalayas International Conference on*. [S.l.], 2011. p. 1–5. Citado na página 77.

VIEIRA, K. et al. Intrusion detection for grid and cloud computing. *IT Professional*, IEEE, v. 12, n. 4, p. 38–43, 2010. Citado na página 15.

VISITORS to Sony PlayStation website at risk of malware infection, Acessado em 5 de julho de 2017 em https://www.sophos.com/en-us/press-office/press-releases/2008/07/playstation.aspx. In: . [S.l.: s.n.], 2017. Citado na página 29.

WANG, X.; SUN, L. Ant algorithm inspired immune intrusion detector generation algorithm. In: IEEE. *Network Computing and Information Security (NCIS), 2011 International Conference on*. [S.l.], 2011. v. 2, p. 124–127. Citado na página 48.

WATKINS, A.; TIMMIS, J.; BOGGESS, L. Artificial immune recognition system (airs): An immune-inspired supervised learning algorithm. *Genetic Programming and Evolvable Machines*, Springer, v. 5, n. 3, p. 291–317, 2004. Citado na página 33.

XIA, Q.; CHEN, T.; XU, W. Cids: Adapting legacy intrusion detection systems to the cloud with hybrid sampling. In: *CIT*. [S.l.: s.n.], 2016. p. 508–515. Citado na página 52.

XING, T. et al. Snortflow: A openflow-based intrusion prevention system in cloud environment. In: IEEE. *Research and Educational Experiment Workshop (GREE), 2013 Second GENI*. [S.l.], 2013. p. 89–92. Citado 6 vezes nas páginas 16, 62, 64, 66, 122 e 123.

XING, T. et al. Sdnips: Enabling software-defined networking based intrusion prevention system in clouds. In: IEEE. *Network and Service Management (CNSM), 2014 10th International Conference on*. [S.l.], 2014. p. 308–311. Citado 2 vezes nas páginas 21 e 51.

YAN, Q.; ZHONG, Y. A radial basis function neural network based on artificial immune systems for remote sensing image classification. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *International Conference on Earth Observation Data Processing and Analysis*. [S.l.], 2008. p. 72850I–72850I. Citado na página 33.

YANG, H. et al. A survey of artificial immune system based intrusion detection. *The Scientific World Journal*, Hindawi Publishing Corporation, v. 2014, 2014. Citado na página 16.

YANG, J. et al. Distributed agents model for intrusion detection based on ais. *Knowledge-based systems*, Elsevier, v. 22, n. 2, p. 115–119, 2009. Citado na página 48.

YE, X. et al. An anomalous behavior detection model in cloud computing. *Tsinghua Science and Technology*, TUP, v. 21, n. 3, p. 322–332, 2016. Citado 4 vezes nas páginas 4, 55, 56 e 57.

YU, J.; WANG, F. Simulation modeling of network intrusion detection based on artificial immune system. In: IEEE. *Software Engineering (WCSE), 2010 Second World Congress on*. [S.l.], 2010. v. 2, p. 145–148. Citado na página 48.

ZARGAR, S. T.; JOSHI, J.; TIPPER, D. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. *IEEE communications surveys & tutorials*, IEEE, v. 15, n. 4, p. 2046–2069, 2013. Citado 2 vezes nas páginas 27 e 28.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, Springer, v. 1, n. 1, p. 7–18, 2010. Citado 2 vezes nas páginas 4 e 26.

UNIVERSIDADE FEDERAL DE SERGIPE
PRÓ-REITORIA DE PÓS-GRADUAÇÃO E PESQUISA
COORDENAÇÃO DE PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Ata da Sessão Solene de Defesa da Dissertação do Curso de Mestrado em Ciência da Computação-UFS.
Candidato: Roberto Vasconcelos Melo

Aos 12 dias do mês de dezembro do ano de dois mil e dezoito, com início às 14h00min, realizou-se na Sala de Seminário do DCOMP da Universidade Federal de Sergipe, na Cidade Universitária Prof. José Aloísio de Campos, a Sessão Pública de Defesa de Dissertação de Mestrado do candidato **Roberto Vasconcelos Melo**, que desenvolveu o trabalho intitulado: *" Abordagem Imunológica de Segurança Baseada em Correlação de Alertas e Redes Programáveis"*, sob a orientação do Prof. Dr. **Douglas Dyllon Jeronimo de Macedo**. A Sessão foi presidida pelo Prof. Dr. **Douglas Dyllon Jeronimo de Macedo** (PROCC/UFS), que após a apresentação da dissertação passou a palavra aos outros membros da Banca Examinadora, Prof. Dr. **Edward David Moreno Ordonez** (PROCC/UFS) e, em seguida, ao Prof. Dr. **André Ricardo Abed Grégio**(UFPR). Após as discussões, a Banca Examinadora reuniu-se e considerou o mestrando (a) APROVADO "*(aprovado/reprovado)*" SEM "*(com/sem)* " ressalvas. Atendidas as exigências da Instrução Normativa 01/2017/PROCC, do Regimento Interno do PROCC (Resolução 67/2014/CONEPE), e da Resolução nº 25/2014/CONEPE que regulamentam a Apresentação e Defesa de Dissertação, e nada mais havendo a tratar, a Banca Examinadora elaborou esta Ata que será assinada pelos seus membros e pelo mestrando.

Cidade Universitária *"Prof. José Aloísio de Campos"*, 12 de dezembro de 2018.

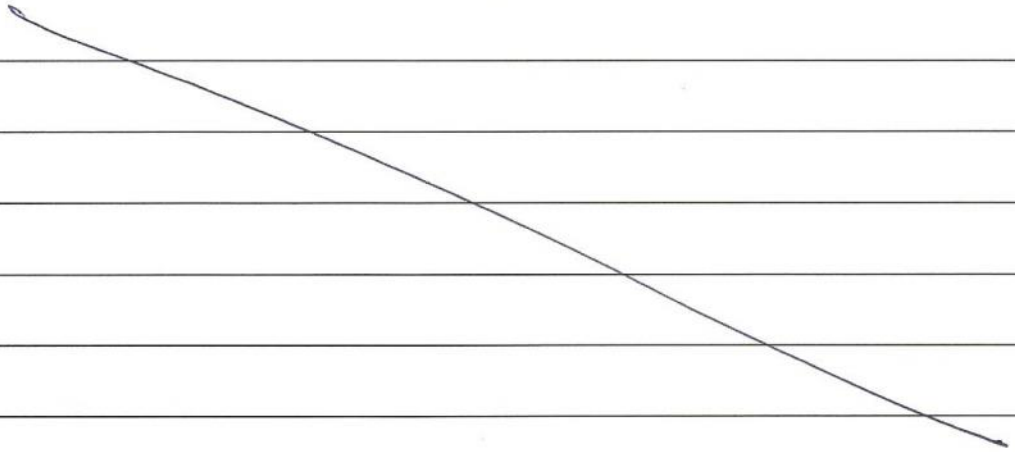
Prof. Dr. Douglas Dyllon Jeronimo de Macedo
(PROCC/UFS)
Presidente

Prof. Dr. Edward David Moreno Ordonez
(PROCC/UFS)
Examinador Interno

Prof. Dr. André Ricardo Abed Grégio
(UFPR)
Examinador Externo

Roberto Vasconcelos Melo
Candidato

Observações (em caso de aprovação com ressalvas):



Prof. Dr. Douglas Dyllon Jeronimo de Macedo
(PROCC/UFS)
Presidente

Prof. Dr. Edward David Moreno Ordonez
(PROCC/UFS)
Examinador Interno

Prof. Dr. André Ricardo Abed Grégio
(UFPR)
Examinador Externo

Roberto Vasconcelos Melo
Roberto Vasconcelos Melo
Candidato